TAPR

ARRL

17th

ARRL and TAPR
DIGITAL
COMMUNICATIONS
CONFERENCE

CHICAGO, ILLINOIS
September 25-27, 1998

17th ARRL and TAPR DIGITAL COMMUNICATIONS CONFERENCE

Co-Hosts:

Chicago Amateur Packet Radio Association (CAPRA)
Packet Radio Users Group of Japan (PRUG)

American Radio Realay League, Inc.
225 Main Street
Newington, CT 06111-1494 USA
tel: 860-594-0200   WWW: http://www.arrl.org/

Tucson Amateur Packet Radio
8987-309 E. Tanque Verde Rd #337
Tucson, Arizona 85749-9399 USA
tel: 940-383-0000   WWW: http://www.tapr.org

Welcome!

As we near the end of the century and the millenium, we face a fascinating future for electronic communication—and it's a future that many believe will be primarily digital. What is the role of Amateur Radio in the digital world to come?

You'll find many of the answers here—at the Seventeenth ARRL and TAPR Digital Communications Conference. Just browse through these proceedings and you'll see what I mean. The Automatic Packet Reporting System is at the forefront of amateur digital activity today and it has already proven itself in critical public-service applications. In these proceedings you'll find no less than *seven* articles on the topic. The other hot subject is spread spectrum, and we have *four* outstanding articles for you to enjoy. You'll also find articles on high-speed packet projects, an evaluation of CLOVER and CLOVER II, and much more.

I encourage you to consider these ideas and take action. We need active, digitally minded amateurs to bridge the gap between the arena of ideas and real-world applications. If Amateur Radio is to remain relevant in the 21st century, we must embrace digital techniques.

David Sumner, K1ZZ
ARRL Executive Vice President

September 1998

Table of Contents

WHATS NEW FOR APRS® IN 1999

Bob Bruninga, WB4APR@amsat.org
115 Old Farm Ct
Glen Burnie, MD 21060

Since APRS was first introduced at the 1992 TAPR/ARRL Digital Communications Conference it has evolved to fulfill a growing need for tactical real-time digital communications. Trying to describe its usefulness is similar to describing amateur radio itself. The scope is so broad and the applications so widespread, that no single listing can be complete. Major milestones in that evolution were the transition from hand entered maps to the USGS CD ROMS, in 1994, the development of MacAPRS in 1994 by Keith and Mark Sproul followed by the official WinAPRS version in 1995. In 1997 Brent Hildebrand developed a special application called APRS+SA to take advantage of the very popular Delorme Street Atlas CD ROM maps. This improvement in maps to the street level was completed recently with the full integration of Precision Maping system into the WinAPRS product in 1998. Along the way, Steve Dimse's javAPRS began the great migration of APRS onto the information super highway in 1996 by making APRS tracking available to anyone with a WEB browser, culminating in his debut of APRServe at the 1997 DCC in Baltimore Maryland which provides a worldwide INTERNET backbone for all APRS packets.

APRS NATIONWIDE FREQUENCIES:

During 1998, APRS accomplished a phenominal natiowide QSY of thousands of users, over 400 digipeaters, and dozens of gateways to a new ARRL and AMSAT sanctioned national frequency. Although there are still some minor areas working on the change, the consistent nationwide channel gives mobile users the freedom to travel without concern for loss of connectivity. Since its introduction, APRS has mainly been used on only two bands, 2 meters VHF and 30 meters HF and mostly on only ONE frequency per band! The fact that thousands of users are all getting so much fun out of just one HF frequency of 10,149.2 KHz and 144.39 MHz is a tribute to the channel efficiency of APRS. In addition, APRS is now beginning to grow onto 6 meters in a system called PROPNET as an easy propogation display tool. Also, APRS sees many applications in Space. APRS has been tested experimentally on several SAREX missions since 1995, the SPRE mission in 1996 and even via Mir in March 1998. The untapped capability on some of the existing Amateur Satellites for efficient APRS communications will also be addressed in this paper.

Today, in 1998, I can say that APRS is becoming a worldwide real time communications system which is revolutionary in Amateur Radio that will again place amateur radio emergency response capability ahead of the leading edge of technology available to the consumer. This paper will describe some brand new developments for 1999 in Internet gateways, digipeating, satellite links and handheld personal APRS comunications devices in addition to what I think is the future direction for APRS into the next century.

INTERNET GATEWAYS:

Steve Dimse's APRServe software has not only revolutionized the long haul distribution of APRS packets, but tied in with Mac/Win/APRS+SA user software, it has turned APRS into a worldwide 2-way messaging system! Any two APRS users worldwide may exchange messages in real time *if* they are both within a

few digipeaters range of an APRS internet Gateway. That is easier than you think. ANY station that is running Mac/Win/APRS+SA automatically serves as a two-way messaging IGATE while he is logged on to the internet. Many people with full time ISP access leave their IGATES running 24 hours a day just to link their RF LAN into the worldwide APRS system. Right now, I just checked and I saw 42 IGATES on line all across the USA. These IGATES linked with over 500 APRS digipeaters provides a worldwide APRS communications system with extrordinary reach.

I keep saying worldwide because we frequently have seen stations logged on from Japan, England and the Neteherlands. It is very impressive to be driving along with your laptop, and communicating across the continent. AND without any routing effort on your part! APRS is the internet of amateur radio. Most APRS stations are not even aware that this worldwide capability exists, because they will not see any IGATEd traffic unless it is addressed to them. Similarly, you cannot send a cross country message unless you know the call of someone to send it to. During Field Day this year, we experimented with allowing CQFD messages to traverse the sytem with excellent results. At least 44 stations made nationwide APRS contacts from their FD sites.

Of course, many people will say, but you need a LAPTOP to send and receive messages and this is cumbersome and usually only worthwhile on long trips where it is worth lugging along the Laptop, TNC, radio and antenna.

But what if all this was combined into a single HT? **READ ON!!!**

APRS DIGIPEATING:

One of the key aspects to APRS that makes it so easy to use is the use of generic digipeating. Thus new users and mobile users may transmit their information into the APRS system without any prior knowledge of the newtwork, paths or routes. Generic digipeating was initially possible with off-the-shelf TNC's but from the beginning I proposed more intelligent routing to avoid duplication and reduce packet overhead. The first new algorithm in this trend was pioneered by PacComm in 1995 which used callsign substitution to help eliminate duplications and to make reverse tracing of paths possible. As a footnote to history, PacComm developed this capability for tracking the location of almost 300 stations in the Bosnian conflict.

Quickly APRS digipeaters across the country began to upgrade to the PacComm 4.0 ROMS and everyone began to see a notable improvement in channel throughput because not only could multiple duplications be eliminated, but also longer generic paths could be used without duplication. Then in 1998, Kantronics finally implemented the long proposed APRS WIDEn-N algorithm. This algorithm provides the same multi-hop long distance routing without duplication, but also eliminates the lengthy explicit digi path that is included in every packet. With WIDEn-N, a single WIDE5-5 digipeater specification replaces WIDE,WIDE,WIDE,WIDE,WIDE in every packet. For each explicit hop eliminated, there is a 7 byte savings per packet. APRS builders in the state of Washington leapfrogged all other networks and became the first state to implement a statewide WIDEn-N digipeating system. Any station in the state may communicate with any other station within 70,000 square miles using only WIDE5-5 as the path and these packets are 28 bytes shorter than before for an approximately 28% improvement and probably a doubling of channel capacity by eliminating dupes!

DIGIPEATING IN 1999: One of my original APRS digipeating ideas is yet to be implemented. This is the -N routing by SSID only. This routing is the same as the WIDEn-N except it dispenses with even the 7 bytes used with WIDEn-N!. Since the "WIDE" in WIDEn-N is itself generic, it can be eliminated compeletely as long as we have some way to indicate the -N number of hops desired. -N routing uses the TOCALL SSID as the routing indicator. Any TOCALL-N will indicate to the network that the packet is to be digipeated N times. Thus we have not only eliminated another 7 Bytes per packet, but opened up even further possibilites. The SSID routing system has been built into all APRS Mic-Encoders in anticipation of the implementation of SSID routing. This algorithm was necessary to keep the Mic-E Bleep as short as possible so that it would be tolerated on voice repeaters. Since hops beyond 7 show a deminishing probability of success, the SSID's of 8 through 15 were reserved for DIRECTIONAL ROUTING according to the following table:

-8	North	-12	North DX
-9	South	-13	South DX
-10	East	-14	East DX
-11	West	-15	West DX

Directional routing is implemented by the sysop for each digipeater who knows the best route for packets from his digipeater to take in each of the general directions. Thus, each digipeater has 4 additional UNPROTO memories, one for each of the cardinal directions. If a packet is received with a -8, for example, the digipeater would substitute the Northern route into the packet before forwarding. The packet would travel this route to the end.

If however, a -12 were used, then the packet would be transmitted by the digi with only the next northern digipeater as one hop, but at that digipeater, again, the -12 would indicate a further NORTH routing. This would be theoretically infinite with the packet always being forwarded NORTH by each digipeater in turn. Eventually collisions will take their toll and the packet would die. But it would have traveled a considerable distance north! This algorithm assumes the same dupe comparator at each hop as is used in the WIDEn-N algorithm so that loops are canceled.
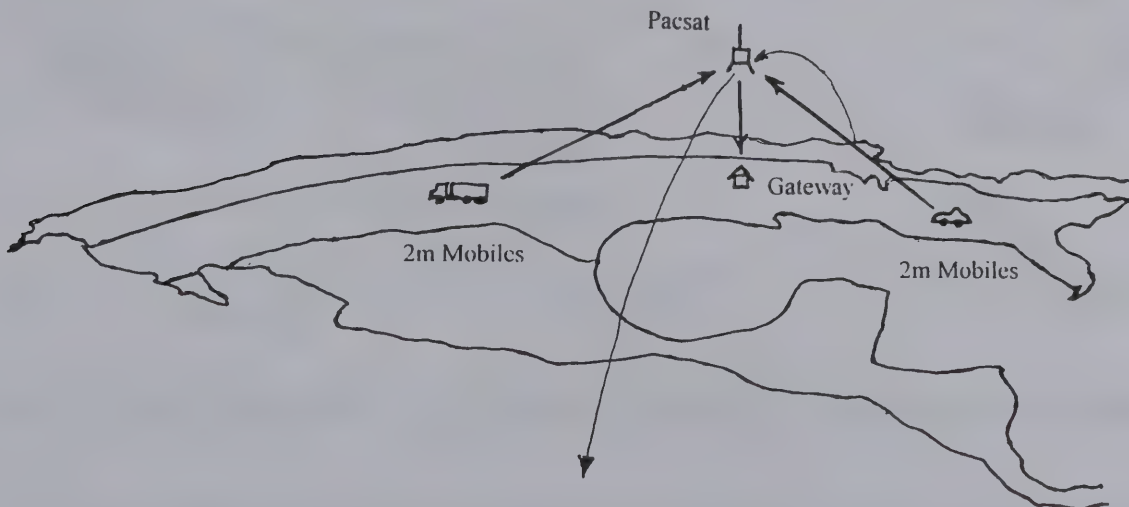
Figure 1. With only one ground station per footprint acting as an Internet Gateway, mobiles over half of the USA can be tracked per pass. Mobiles transmit using conventional 2 meter FM mobile radios and standard AX.25 APRS packets. A slight ($3) mod to any TNC can configure it for the required manchester uplink of the 1200 baud Pacsats.

APRS MOBILE SATELLITE NETWORK:

APRS has been demonstrated several times via the Space Shuttle and SPRE experiments as a very efficient way for many stations to share a single space-based digipeater. In 1998, a nationwide school test was conducted using APRS via the Space Station MIR [1]. With the short duration of APRS packets and the fact that each station only needs to successfully get one packet through to convey his position and status, APRS is ideal for allowing the maximum number of stations to participate per satellite pass.

Fortunately, there are already a few 1200 baud AX.25 2 meter FM satellites in orbit! We currently have authorization to trasmit APRS packets via AO-16 and possibly others once we demosntrate the usefulness of these satellites to serve the mobile amateur radio operator. All it takes is any 2m FM mobile rig, your normal TNC (with a $3 mod) and your normal mobile antenna. With 25 watts you can hit the bird a few times a day while you are on long distance travels [2].

Unfortunately it is not quite as easy to hear the satellite. But ONLY a few downlink stations are necessary to feed the downlinked packets into the worldwide internet linked distribution system to have your packets arrive at their intended destination. Currenlty there are several stations working on automating the downlink and Steve Bible is working on making the $3 XOR gate mod readily available for experimenters... For further information, see my TRAKNET paper in the AMSAT PROCEEDINGS [3]

Figure 2. This is an APRS screen capture after the third pass of MIR during the special APRS-MIR tests conducted on 11 March 1998. It demonstrated the potential of using an amateur satellite digipeater for relaying mobile position/status/messages.
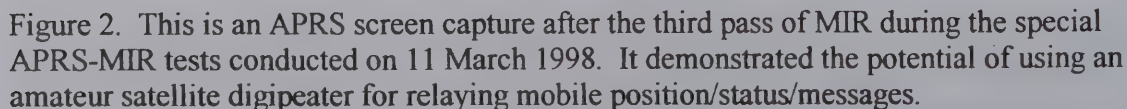
GREATER APRS CHANNEL CAPACITY FOR 1999:

Since APRS was first introduced, we have been striving to improve channel efficiency where ever possible. Currently, an evening's monitoring in the Baltimore area will capture about 200 stations within about a 200 mile radius. This is possible because of the great strides that have been made with digipeaters that has almost tripled channel capacity by reducing the number of bytes in each packet and by eliminating all the redundant duplication of each packet. However, there is still plenty of excess verbage in most packets.

The Mic-E lead the way in 1997 with the shortest possible position report of only 8 bytes. Since then we have been wanting to introduce a compressed algorithm that would not only shorten the usual 26 byte position to a much shorter packet, but also allow for additional precision to meet the potential of DGPS systems. In 1999, you will begin to see these packets. For example the following packet can be compressed to only 13 bytes:

Normal: WB4APR>APRS,WIDE*:=3859.11N/07629.11Wv123/045...

Compressed: WB4APR>APRS,WIDE*:=/YYYYXXXXvcsT...

Where the YYYYXXXX contains the LAT/LONG to the nearest foot worldwide and the csT contains both the course and speed. The T is a TYPE byte that indicates what format was used in the csT bytes. Notice that the 7 bytes after the position report in an APRS packet is a field that is used for several mutually exclusive purposes. These are all encoded in the csT bytes.

CSE/SPD	This is for moving stations
PHGxxxx	This is for stationary stations
DIR/SPD	This is for Weather stations
ccccccc	Additional coment field if needed
/A=123456	For GGA packets the altitude can also be compressed

Thus the compression algorithm saves us another 13 bytes as high as 37%. Or looking at it another way, this allows 37% more users on the channel. All versions of APRSdos since 820 are compatible with this protocol. Unfortunately all versions prior to 820 had an earlier draft compressed protocol that was subsequently modified. Thus we cannot begin to transmit the new protocol until ALL APRS stations upgrade to 820 or later. The compressed format will be an option since it does compromise in one area. It rounds COURSE to +/- 2 degrees and speed to +/- 2% or so.

ON-THE-FLY DIGIPEATER COMPRESSION:

At first it may appear that the advantages of compression are small since they will only apply to stations running APRS PC software. But this is not the case. Almost ALL stations may use the compressed algorithm as follows:

DIIGPEATERS:	The compressed format is entered into the BText
DOS/Mac/Win/+SA:	A selectible option for any position or object
KPC-3 TRACKERS:	Kantronics has signed up to generate the compressed algorithm directly for a 300% shortening of NMEA!
OTHER TRACKERS:	The NMEA is converted at the digipeater! This will also be built into KPC-3+ digipeaters!

Notice that some of the longest packets on the air are the raw NMEA strings from GPS trackers. These packets may be compressed 300% or so using the compression algorithm. But not only will this occur for new KPC-3+ trackers, but also any KPC-3+ that is used as a digipeater can compress any RAW NMEA that it hears into the compressed format before forwarding!

For backward compatibility, or if you do not want your packets compressed by the network, simply continue to use the default TOCALL of GPS and these packets will not be compressed. But trackers that use the new GPSxyz format for indicating their xyz ICON will be automatically compressed at the digipeater. If a station does not want to be compressed, he should use the equivalent SYMxyz or any other valid APRS TOCALL.

KENWOOD APRS HANDHELD COMMUNICATOR:

And now, what you have all been waiting for! Kenwood is just now announcing at DCC as you read this, the production of their Personal Data Communicator that has APRS FULLY INTEGRATED INSIDE! Here are the features:

Dual Band 5 watt HT
Built in TNC
Built in GPS modes (with external GPS)
Built in APRS displays
Built in APRS messaging
Built in APRS Mic-Encoder
Built in DX cluster Spotting

Need I say more?

Here is how you may use your Kenwood APRS Data Communicator

1) Plug in your laptop and operate normal packet.
2) Use your laptop and the HT for fully portable APRS.
3) Plug in a GPS and operate as a stand alone tracker.
4) Plug in a GPS and operate voice with your POSIT and comment going out in a Mic-E burst on the end of your transmissions.
5) Operate simultaneous voice and APRS. Voice on either band, while your APRS packets go out on 144.39
6) Unplug everything and just use the built-in APRS displays! The radio will capture positions, status, bulletins, WX warnings DX cluster spots and personal messages!
7) Unfortunately, the new HT cannot digipeat.

APRS MESSAGES: The fact that the new HT can both send and receive APRS messages opens up APRS to become a worldwide amateur 2-way personal messaging system! Althought the primary design goal of the new HT was the GPS interface for position reporting, it was only during BETA testing that we noticed that we were using the messaging system ALL THE TIME to coordinate our testing. What we soon realized was that although the GPS interface is the pre-emminant feature of the new HT, most of the time, however, a HAM is not going to bother with integrating the GPS for casual routine operations. BUT he WILL use the messaging capability since it requires no attachments!

Larger than life. Actual size 4.8 by 2.3 inches.

You may think that messaging is no big deal, since you can simply talk into the radio, but that is what is so amazing about the APRS message capability. To talk to someone on voice, all of the following conditions must be met:

    You both must have an agreed freqency or repeater
    you both must be on the radio at the same time
    The frequency or repeater must not be in use by others
    You are limited to a communication range of about 20 miles
        (the range of one repeater. Or further on some linked systems)
    You have to continuously monitor the speaker to hear a call

Now compare this with APRS messaging in the   new handheld HT ,   All you do is enter the message! The only conditions that must be met are:

    You are both within range of an APRS digi/igate ANYWHERE IN THE WORLD
    Your radios are on.

Notice that no prior knowldege is necessary to communicate. Just enter the call and the message. IF the other station is on APRS, he will get it. Have you ever arrived in a new town and have no idea how to contact other area HAMS because you didnt agree on a time, a frequency and a plan? With APRS, just eneter the message, and it will get through! (most of the time...)

APRS WORLDWIDE E-MAIL: The use of the new HT for messaging is limited only by the imagination of the APRS developers. The year 1999 will see the introduction of APRS E-Mail engines that can convert APRS messages to and from Email from any user in the country.

APRS Data HT DISPLAYS: Obviously the display on the new HT is quite small, so it cannot display as much as a full APRS LAPTOP. But KENWOOD engineer Shin Aota has done an excellent job of capturing the most essential information. Here are some of the displays:

STATION LIST:	11:WB4APR-15 12:WU2Z 13:K4HG-4	Maintains a list of the last 40
POSITION COMMENT:	11:WB4APR-15 /comment. text 20 c	Displays 20 characters
POSITION GRAPHICS:	12:WB4APR-15 FM19QD 10.6mi	Shows ICON, Grid SQ, Distance and direction!
POSITION:	13:WB4APR-15 N 39 09.48 W 076 33.23	
COURSE/SPEED:	14:WB4APR-15 cse000 sp000	Course and speed if moving
POWER HEIGHT GAIN:	19:WB4APR-15 pwr50w h 80' ant3db omni	Or PHG if entered

```
WEATHER:        │19:WB4APR-15│   Or weather if available
                │dir000 s000m│
                │t 89 f r000"│
```

Notice how the memory location of WB4APR-15 began here as location 11 but
slowly got bumped down in the list as time progresses. New stations always
appear in location 1, and all others move downward. When WB4APR transmits
again, he will go back to the top of the list. Thus you get a good idea of
the age of the packet... All of the above screens are extensions to the
POSITION/STATUS memory list.

There is a separate MESSAGE list that captures all Bulletins, NWS warnings,
and Messages. Up two 16 messages are retained and each message has a two
screen display. The first 24 characters of the message are shown on screen
one and the remaining 20 are on screen two. Memory is limited to only 45
characters per message. This is in contrast to the normal APRS message
protocol that can contain 67 bytes. But this is easily handled as noted
below:

```
BULLETINS:      │A<WB4APR-15 │   Bulletin A from WB4APR-15
                │This is page│
                │1 of two pag│   up to 45 characters

                │A<WB4APR-15 │   Screen two
                │es that can │
                │be used.    │

MESSAGES:       │M>WB4APR-1 1│   Incoming message to WB4APR
                │Notice there│   Notice the LINE number 1
                │are 24 chars│   in the upper right corner.

                │M>WB4APR-15 │
                │on 1st page&│   Screen two holds only 20 chars
                │20 on 2.    │
```

Although the message display is somewhat disjoint, the other APRS authors are
integrating these limitations into their software. In APRSdos, for example,
you will notice two faint Gray lines in the SEND MESSAGE BOX that mark the
locations of the PAGE break and maximum line length for a Data HT message.
Thus users can easily see how their message will appear on receipt and know
when to stop typing at 45 characters. Similar gray lines will show users the
limits on the INPUT-MY-STATUS and INPUT-MY-POSITION
comment prompts.

Whenever a new packet comes in that has not been heard before, a NEW PACKET
display pops up to show the latest packet:

```
                │WB4APR-15   │   New Packet Display
                │  Text of ne│
                │S w packet..│   The letter S shows it was a STATUS
```

If a packet comes in that is a dupe of a previously held packet it displays
only a single line and indicator of the type of packet in the lower left
corner:

```
                │ 144.390   D│   Dual band freq showing data chnl
                │>445.925    │   > shows voice channel
                │dP WB4APR   │   This indicates a duplicate posit
```

POSITION LIMIT: Early on, it was a primary consideration how to limit the number of stations captured, since the memory could only hold 40 stations. This was easily accomplished using the range function that is displayed with every packet. The user can set a POS LIMIT in miles. Thus, any position packet beyond this distance limit will be ignored. The duplicate data display will show >P WB4APR for any station that is so ignored. This is a very powerful capability and allows the user to limit his memory and displays to only his area of interest. Thus, stations may work a public service event and not have their screens cluttered by other stations on frequency that are out of area!

ALTNETS: In addition to the POS LIMIT, the new HT fully implements the APRS ALTNET and TOCALL filters. All stations at special events should set their TOCALL to SPCL. With SPCL, they will only capture and display other stations using the call of SPCL. But all other monitoring stations will still see them. The ALTNET concept allows special subgroups of operators to select unique ALTNET calls and then only they will see each other. This is useful when a small group of stations are conducting experiments on the main APRS frequency and do not want to clutter everyone else's displays with their traffic.

QUERIES: Not only does the POS LIMIT establish your receive limit (0 to 5000 miles) but it is also your QUERY range. If you send a Query, the Query will trigger a response only from stations within that range of your position. [Queries were not in the prototype and may not make it in time for production]

ICONS: Although the ROM could only support 15 ICONS, these were carefully chosen to handle the majority of cases. Any of the 350 or more APRS ICONS may be transmitted and received, but only these 15 will show as actual graphics ICONS. The rest will display only the two character equivalents. Those marked with an * can display an overlay character.

Kenwood	SSTV	Triangle*
Jogger	Airplane*	Jeep
House	Boat*	RV
Portable (tent)	Car*	Pickup Truck*
Sailboat	Bicycle	Van
Digi*	Gate*	WX*

DISCLAIMER: Since the new HT is still in beta test at the time of this writing, nothing in this paper should be construed to describe the final product.

Be assured, that the built-in GPS tracking and handheld two-way messaging capability of the new HT combined with the worldwide APRS infrastructure will make

1999, the YEAR OF APRS!

References:

1] **An APRS/Mir Test**, Bob Bruninga, The AMSAT Journal, May/June 1998.

2] **Proposal: An AMSAT Mobile TRAKNET**, Bob Bruninga, The AMSAT Journal, May/June 1997.

3] **TRAKNET, An AMSAT Mobile Satellite System**, Proceedings of the AMSAT-NA conference, 16-18 Oct 1998 Vicksburg, MS.

Internet to RF Messaging within APRS

Steve Dimse, K4HG
189 Le Grand Lane
Cudjoe Key, FL 33042
k4hg@tapr.org

Abstract

This paper describes a new feature implemented within the APRS system, which allows transparent messaging between stations on the Internet and RF sides of the network, as well as on two distant RF networks using the Internet as a link.

Introduction

In a paper presented at the Digital Communication Conference in 1997 I described the genesis of the Internet portion of the Automatic Position Reporting System (APRS) network. This is done with the APRServe software, which serves as the hub for interconnection of APRS users around the world. My work on APRServe has continued, and this paper will detail the Internet to RF messaging system made operational over the last year.

APRS remains one of the most vibrant and exciting aspects of amateur radio. Beginning on HF and VHF, the system has expanded to include an Internet portion, which enables the display of a thousand or more APRS stations nationwide. This is accomplished with a linked backbone of servers running APRServe software, and Internet Gateways, or IGates, which pass data to the backbone. Until recently, these IGates were one way—data could get from the RF network to the Internet, but there was no way for information to flow in the opposite direction.

With an addition to the APRS protocol and modifications to the Internet gateway software and the client programs, it is now possible to send a message from the Internet to a station on RF. Furthermore, it is possible to send a message between stations on two different VHF networks. This is done transparently to the user.

Issues

There are two problems that a bi-directional link between RF and the Internet could introduce into the APRS network. First, the volume of traffic handled by APRServe could easily overwhelm the capacity of the 1200-baud VHF network and the 300-baud HF network. Second, in order to ensure compliance with FCC regulations, the IGates must be certain that a message to be passed onto RF was originated by a licensed amateur.

Traffic Control

The Internet has vastly greater bandwidth than the RF network, making it necessary to very carefully limit data coming through the gateway. In order to do this, each IGate keeps a list of "local" stations, where local is defined as stations being within a certain number of hops. The optimum number of hops will vary based on specific local network configurations, but generally 2 or 3 is used. Only messages destined for "local" stations are transmitted by an IGate. The system also does not pass messages that are both from and to a local station, which prevents local RF QSOs from being echoed from the Internet.

Since the HF network consists of a single 300-baud channel that is shared between all users in the country, no messages are gated from the Internet to HF.

Validation Protocol

When initially implemented, APRServe accepted any data that was passed to it, acting like a gigantic party line. However, in order to comply with FCC regulations it is necessary to make certain that only hams can initiate a transmission. The Sprouls and I worked out a method to do this with as little impact to the user as possible. All APRS client programs now send a logon message, which identifies the callsign of the user, the version of the program, and optionally includes a validation number. Based on this message, APRS places each connection into one of three classes. Read only access is granted to those connections, which do not send a logon message. This can occur with obsolete version of the software and with telnet clients. Internet-only access goes to those users who have sent a logon message without a validation number. This class of connection allows users to send their own position reports and messages to internet users, but does not allow any of this data to be sent over RF. Each packet from such a user is identified as untrusted, and every IGate checks to be sure a packet is trusted before sending it on RF. Finally, a logon message which includes a validation number which matches the callsign provides a user with full access.

APRS Protocol Addition

While a full discussion of the APRS protocol is beyond the scope of this paper, an understanding of the system in general is necessary to understand how the messaging works. Unlike other packet radio systems, APRS utilizes the unconnected format within AX.25. While unconnected packets are well known to be inefficient for transmission of data from one station to another, APRS uses them for the dissemination of data from one station to many. This is far more efficient than creating a connection between each APRS user within an area.

When the APRS protocol was first designed, no provision was made for a station to forward data from another station (other than the standard AX.25 digipeating protocol). We added a packet format we call "third party" to handle this situation. It is important to note that this is not the same as the meaning as "Third Party" used in the FCC rules, which refers to messages from non-hams.

In keeping with the other APRS formats, which stress human readability, the other APRS authors and myself chose to simply allow the inclusion of any legal APRS packet in the form it comes out of the TNC while in monitor mode. For example, the following is a typical message packet as seen on the APRS network:

```
K4HG-5>APRSM,WIDE:WU2Z      :Hi Keith{1
```

The new protocol simply takes this line, prefixes it with the character '}', and retransmits it, appearing like this to a monitoring station:

```
KD4DDO-2>APRS,WIDE:}K4HG-5>APRSM,WIDE:WU2Z      :Hi Keith{1
```

By using this simple method of encapsulation, we have enabled the retransmission of any APRS packet. This also turns out to be simple to parse, since it is only necessary to send the text following the '}' recursively to the parser. In practice, we also modify the path info to be more descriptive of how the packet got retransmitted, so it would become:

```
KD4DDO-2>APRS,WIDE:}K4HG-5>APRSM,TCPIP,KD4DDO-2*:WU2Z      :Hi Keith{1
```

Position Reports

In order to avoid overloading local area RF networks, we have not allowed the IGating of position reports. The thousand or more position reports seen on the Internet would overwhelm even the emptiest RF network. However, since a user involved in a QSO with another ham would like to see that person's position on the map, an IGate sends a position report once every 15 minutes.

Prediction

As is my custom I will close with a prediction of what I will be presenting at next year's DCC. I expect to expand this messaging system to include email, so that a user anywhere on an APRS network will be able to send short email messages to anyone in the world. This should greatly increase the utility of APRS in disaster communications.

Conclusions

APRS now has of seamless integration of its RF and Internet networks. This produces a much more capable and robust system, and has some interesting uses in emergency communications. It is now possible to set up a link in a disaster area using APRS, and provide Internet messaging to all users within that area.

CO-EVOLUTION of PRINT, COMPUTER, and RADIO TECHNOLOGIES

by Roy Ekberg, WØLIQ and Martin Schroedel, K9LTL

ABSTRACT

This is about standards for CAC (Computer Assisted Communication) proposed first in the 12th and 16th DCC proceedings. Our CAC Model 17 was entered in ARRL's library under C-135-1. R & D work began in 1989 and continues.

KEY WORDS/PHRASES

Categorization, inverted communication, status-line, Digital shorthand, and Digitex.

WHY PRINT-COMPUTER DATABASES CAN BE CORRELATED VIA CATEGORIES

John M. Ellis, Professor of German Literature at the University of California at Santa Cruz, outlined his views on *categorization aspects*. He related these to communication and linguistic functions as follows:

> Categorization...is the most basic aspect of language, and it is a process that must be understood correctly if anything else (including syntax) is to be understood; and categorization, not communication, is the most important function of language; one that is prior to all others.[1] JME, pg. 27

> For communication to be possible, then, there must first have been a considerable degree of processing experience—of analyzing it, abstracting from it, focusing and shaping it. It is in this complex process that the essence of language is to be found, not in communication per se. Indeed, communication is only of value to us because this prior process creates something that can be communicated and that is worth communicating.[1] JME, pg. 29

Linguistic categories let hams use multi-page, manual, randomly accessible, print media with single-paged, powered, randomly accessible, P/Cs. This convergence is analogous in part to using telephone directories (print media) to utilize computerized telephone switching systems. CAC's categories are: database, files, subjects, indexed records, and novel complements. The first four are *managed* in CODE mode; novel category is *managed* in TYPE mode (as illustrated later in the text).

INVERTED AND REGULAR COMMUNICATIONS NEED UNIVERSAL STANDARDS

Inverted communications are types in which data at receiving hams' QTHs are managed via hams who transmit control signals. Regular communications must be used also to include unrecorded *novel* items, to perform CQing activities, etc. Universal CAC system standards are required which will ensure that CAC system will be practical for international communications in foreign languages.

Digital shorthand includes ++ signs for switching from CODE to TYPE mode or vice-versa. Speech equivalents are CODE-CODE and TYPE-TYPE. (Note: A Morse code for the + sign has never been standardized; we proposed one later in the text.)

CATEGORIZED, FOUR LEVEL, HIERARCHY OF INDEXED RECORDS

The contents of data recorded for CAC system's communications is categorized under a four-level hierarchy: Databases, Files, Subjects, and Indexed Records. Records are referenced via their indexes (i.e., Record Indexes or RIs). Indexes will range from 0 through 9999 per file. RIs are listed under Subjects in files that have encoded names (codes simplify monitoring and switching). CAC users can peruse printed files to select records they want to display at their receiving ham's QTH. Indexes are not memorized! Receiving hams input shorthand strings in P/Cs to display decrypted messages on monitors and/or printers. Status-lines keep users posted regarding their activities in cyberspace and in real time.

UNIVERSAL DIGITAL SHORTHAND SCRIPT AND SIGNALS

Expression *Digital shorthand* replaces "Keypad Interface Language" promoted in the earlier DCC proceedings. The reason is, term *shorthand* suggests the functions of encoded strings more precisely than "language." Keypad keys are still favored as the logical source for universal script in Digital shorthand as follows:

SCRIPT	SSB	CW	FUNCTIONS OF DIGITAL SHORTHAND
0	ZERO	-----	Used for Record Indexes
1	ONE	.----	" " " "
2	TWO	..---	" " " "
3	THREE	...--	" " " "
4	FOUR	....-	" " " "
5	FIVE	.....	" " " "
6	SIX	-....	" " " "
7	SEVEN	--...	" " " "
8	EIGHT	---..	" " " "
9	NINE	----.	" " " "
.	DOT	.-.-.-	Executes DS strings.
-	JOIN	-....-	Joins Record Indexes.
*	CHAT	.-..-.	Converts TYPE mode to CHAT mode.
++	MODE–MODE	---... repeat	Changes CODE mode to TYPE mode, and vice-versa.
//	FILE–FILE	-..-. repeat	Switches from one file to another... via adding file code number.
(	ENTER	-.--.-	Deletes input errors in CODE mode. Adds blank lines in TYPE mode.

SYNONYM KEYS:
Finland, Germany, and Sweden prefer synonyms (X) for (*) and (÷) for (/) on keys. Synonym keys have equivalent functions.

EXPERIMENTAL STANDARDS FOR PRINT AND COMPUTER DATA CATEGORIES

Present computer monitors have 25 lines of 80 characters per line. We presume that future digital monitors will have 120 characters per line based upon the news sources which predict that digital TV monitors will be made 30% wider on screens. We set "Indexed Record" lengths at 40 characters per line (which is half of the present line width). This seems likely to be satisfactory for the anticipated new digital line widths. Forty character line widths are adequate for text because records can be linked to form paragraphs as necessary. Wider screen widths would permit "split-screen" applications on which text could be 40 characters wide (on the left side), and graphics could be 80 characters wide (on the right). Text could be managed using Digital shorthand strings whereas graphics might be managed via other signals suitable for cross-referencing graphics. (We presume that present text width standards will not be obsoleted by the arrival of future hardware.)

CAC system's *status-line* is essential for switching among "virtual categories" (like files) in real time. Status-line posts CAC's categories in an order read left-to-right (likely to be referred to the most often during communications). Modal type changes (from TYPE to CODE, etc.) are made frequently. So, the left-most words on the status-line will post modal changes each time that you will press your ++ keys. The next most monitored category is that of the three-character file codes. It is imperative to know which file that one is managing. Both sending and receiving operators must keep synchronized to communicate. Record indexes will appear above the status-line up until the time of their execution by the DOT command. Then RI's contents will replace the "spent" RIs (which also removes a source of noise). The database category seldom changes, so it is centered on the status-line.

INTERNATIONAL MORSE CODE REMAINS ESSENTIAL TO USE CAC SYSTEM

Opinions are offered in QST that "Morse became obsolete!" That claim is untrue whether CAC system were to become legalized or not. Morse code has been deciphered and displayed on microprocessor digital readout devices for some time. We should appreciate our homo sapien's ability to learn, remember, and copy Morse signals! A psychological copying barrier is often experienced around 8 WPM. Since CAC system can download text in a flash, 13 WPM code requirements could be reduced to 8 WPM. However, hams who enjoy breaking this code barrier would still enjoy using their Morse expertiSe when using TYPE mode in CAC system (in which plain language is used to supplement traffic in CODE mode). Let's not forget that deaf hams can use Morse to their advantage also. People who argue that Morse is "old fashioned" anymore are not likely to claim that Roman letters are "old fashioned" despite the fact that they are over two thousand years old. Functional utility factors must be weighed.

SIGNIFICANT BENEFITS OF CAC'S INVERTED COMMUNICATION SYSTEM

Q-codes serve as an analogue of what is meant by "inverted" communication systems. Upon speaking or sending "QSY" in Morse, most hams will remember that QSY means: "Change to another frequency." QSY, if sent in plain English, requires 28 characters to be coded. Q-codes still have a useful place in CAC system only they will be indexed numerically under a subject category such as "02. Q-CODES." Also, Q-codes would be listed in alphabetical order with three digit length indexes put in numerical order. Even though numeric indexes will be sent to reference Q-codes recorded in memory, only Q-code referents will display after commands execute the displays. Q-codes represent a tiny fraction of useful "message parts" which could be recorded in computer memory for convenient referencing.

Q-codes cannot be accessed conveniently when used without computer assistance. Nor can Q-codes be integrated easily with other recorded message parts unless they will have been assigned arbitrary indexes. Think of computer assistance as a process in which computer's memory is made available to support the user's memory. In this inversion process...transmitting hams lookup things in print media that they wish to communicate. Then, they send signals which receiving hams copy directly (or from decoding devices) and input decoded signals on keyboards. Computers display the messages without taxing any receiving ham's memory. In other words, the burden of communicating useful messages rests upon the operating knowledge and skills of hams who send. Magical displays can be created by sending at only 8 WPM!

Meanwhile, one of the most impressive benefits (unavailable on the Internet) is that sending hams will lookup message parts in any of twenty European languages and use universal Digital shorthand to reference foreign message parts. This is a dream that has come true fostered by philosophers centuries ago (who lacked means)[2].

EXAMPLE OF SUBJECT IN PRINT AND COMPUTER FILE "ALPHA 1" (Coded AL1)

Expression *Digital shorthand strings* is shortened to word *Digitex* when naming strings of written script. Subject 06. in File AL1, database HAMRAD98, follows:

```
06. MAKING CONTACTS: 165-200
165 Thanks for answering my call!
167 I copy you solid!
169 My CALL sign is_
171 Try calling me again in 15 minutes.
173 This frequency is in use!
175 Can you QSY up 2 KHz?
177 Can you QSY down 2 KHz?
179 QSY to frequency (in MHz)_
181 I'll QSY per your request.
183 Please repeat your last transmission.
```

DIGITEX TRANSMISSIONS

Hams will use protocols during CQing operations similar to those used during contests and field day operations. (Those are beyond the scope of this paper.) If TYPE and CODE mode signals were displayed on Morse decoders, they will appear somewhat as in the example listed below:

Five Line Digitex String
K9LTL DE WØLIQ ++165.173.175.++GA K

K9LTL will have inputted this Digitex message on his Morse decoder to read it and copy it on his computer keyboard. The translated display will be as follows:

K9LTL DE WØLIQ
Thanks for answering my call!
This frequency is in use!
Can you QSY up 2 KHz?
++GA WØLIQ

Both operators will be monitoring their status lines to keep track of their virtual category dynamics as follows:

a)
| TYPE   AL1   HAMRAD98   Esc => MENU   F1 => HELP |

b)
| CODE   AL1   HAMRAD98   Esc => MENU   F1 => HELP |

Status line a) confirms that TYPE mode, File AL1, and database HAMRAD98 are active. Line b) confirms that CODE mode replaced TYPE mode. Esc or F1 keys can be pressed to see options and return to the same screen display in process. The sending operator could have sent Digitex to display only four lines by joining record indexes:

Four Line Digitex String
K9LTL DE WØLIQ ++165.173-10-175-.++GA K

K9LTL DE WØLIQ
Thanks for answering my call!
This frequency is in use!_Can you QSY up 2 KHz?
++GA WØLIQ

The additional RI "10" inserts an "underline space" between joined sentences. Color monitors let users read white letters on blue backgrounds. Status lines have yellow backgrounds and black letters. Besides having the option of printed outputs, users could add digitized speech software and hardware to hear spoken messages. Notice that only a few Morse signals are required to display considerable information.

CONCLUSIONS

The authors realize that they are only in the "Kittyhawk" stage of development. Unfortunately, we lack the skills, legal assistance, and funds to promote CAC system technology as much as we would like. We have concluded that Morse isn't obsolete as some hams have claimed! Actually, we need to standardize some more characters.

REFERENCES
(1) Ellis, John M. "LANGUAGE, THOUGHT, AND LOGIC." Northwestern University Press, Copr. 1993
(2) Knowlson, James, "Universal language schemes in England and France 1600-1800." University of Toronto Press, Toronto and Buffalo, Copr. 1975

Optimized Channel Access Mechanisms for Decentralized Spread Spectrum Packet Networks

Matthew Ettus, N2MJI
1200 Dale Avenue #17
Mountain View, CA 94040
(650) 962-8811, ettus@earthlink.net
http://home.earthlink.net/~ettus

Abstract

In a Spread Spectrum (SS) network with cooperative, minimum-energy routing, latency grows with the number of stations in the network due to the increasing number of hops which a packet must take en route to its destination. This is the principal factor limiting the number of stations in the network. Previously proposed channel access mechanisms for these networks were principally concerned with collision avoidance, to the detriment of latency, especially in light or moderately loaded networks. Additionally, while avoiding extremely low SNRs due to collisions, better results are obtainable for average SNR. Several new channel access mechanisms are proposed and simulated for the purpose of addressing the issues stated above. The results obtained indicate that there is potential for significant improvement in performance through the use of alternative channel access procedures.

Keywords

Spread spectrum, channel access mechanisms.

Introduction

In [Shep95] it was shown that cooperatively routed, spread spectrum packet radio networks can scale arbitrarily large, as long as the system uses automatic power control and minimum transmitted energy routing. Such as system will maintain a reasonable signal to noise ratio (SNR) even with huge numbers of stations. Unfortunately, such a network will suffer from latency problems. This is due to the fact that the number of hops (transmissions) a packet must make when traversing the network grows as a function of the number of stations in the network. This is the principal obstacle to large-scale networks of this type.

The total latency is determined by the number of hops taken and the average time per hop. The work in [Ettus97] was aimed at improving the latency situation by reducing the number of hops and minimizing congestion through alternative routing methods.

The motivation for the research described herein is to reduce the time which a packet spends waiting to be transmitted, while at the same time improving the SNR of the system as a whole. To this end, various channel access mechanisms have been investigated.

Medium Access Control

A multiple access system (i.e. TDMA, FDMA, SSMA, etc.) is used to divide up a combined bandwidth, and provide for the separation of signals from each other. The purpose of medium access control (MAC) protocols is to arbitrate and coordinate the usage of the shared channel. There are many MAC protocols which have been developed for traditional, non spread spectrum, packet networks. Commonly used ones include ALOHA, carrier sense multiple access (CSMA), and round-robin (token passing).

In ALOHA, a station may transmit whenever it has traffic to send. This works well in lightly loaded networks, and has minimal latency. In a heavily loaded network it suffers from excessive collisions, resulting in decreasing throughput with increasing load. CSMA (which is used by AX.25 networks) was invented to solve these problems. Before transmitting, a station makes sure that the channel is clear. If it is not, it waits a random time (exponential backoff), and tries again.

In the SS network investigated, we are using spread spectrum multiple access (SSMA). Stations are able to differentiate between multiple simultaneous transmissions through the use of orthogonal spreading sequences.

Many SS networks use some variation of an ALOHA MAC protocol. This has the advantage of low latency, and the use of SSMA reduces the effect of what would be collisions to a reduction in SNR. Under high loads, however, this can be a large loss in SNR, and that causes packet loss if the processing gain (PG) is not increased. A corresponding decrease in throughput occurs.

In order to avoid collisions, a localized coordination system was used in [Shep95]. This was based on a slotted system in which stations were assigned transmit and receive windows based on a pseudorandom hash function. It ensures that a station will not transmit when the desired recipient might also be transmitting, or when a nearby station might be listening. The advantage of this method is that all local collisions (those that have the greatest effect on SNR) can be avoided.

Although this system works very well, especially under heavy loads, it is not perfect. The principal disadvantage is that it is not an adaptive system. If no other stations in the network are transmitting, a station will wait until an appropriate time slot comes along anyway. On the other end of the spectrum, in a very heavily loaded network, even without local collisions, enough distant stations might be transmitting to cause the channel noise level to be high. This causes the system designer to provide enough extra processing gain to overcome these lower SNRs, even though they only occur during heavy load.

Overload-signal spread spectrum (OSSS) [OO98], which provided the inspiration for this research, attempts to solve this problem. Unfortunately, its use only applies to cellular networks (all transmissions to or from base stations). It relies upon the base station to sense its received noise level. When it gets too high, it sends out an "overload signal" which tells the other units to hold off on transmissions. This works best with geographically small networks.

The main thrust of this research was to investigate new MACs which would combine the desirable properties of ALOHA, CSMA, OSSS, and Tim Shepard's slotted system (TSSS for lack of a better name).

Alternative MAC Protocols

A total of five different MAC protocols were investigated. These were ALOHA, TSSS, and three new ones, discussed below. ALOHA allows a station to transmit when it wants to, as long as it is not *already* receiving. TSSS allows a station to transmit when:

- it is in a transmit window
- the receiver is in a receive window
- there are no nearby stations in a receive window which would be stepped on

The new systems all attempt to provide throttling of transmission when the packet probably would not get through. It is impossible to implement CSMA with spread spectrum, as there are too many signals (carriers) to sense. Instead, we use noise sensed multiple access (NSMA). This is essentially an ALOHA system which will not transmit if local noise is high. So long as the noise level at the transmitter (which does the sensing), and the receiver are similar, this will prevent transmission when it would result in poor SNR on reception.

The second new system is basically the same as NSMA, but implements exponential backoff. This is designed to prevent large numbers of stations from jumping back on the channel at the same time after a noisy condition ends.

The last MAC proposed uses NSMA and TSSS. Since it is still slotted like TSSS, this will not directly improve latency. If it improves SNR, however, throughput is improved, and fewer retries are necessary.

Simulation

In order to gain a better understanding of how these various channel access mechanisms affect the performance of the network, it was necessary to perform simulations. A simulator, based on SSNetSim from [Ettus98], was developed for this purpose. The hash-based coordination function had been previously implemented, but the three noise-based functions needed to be created.

For the purposes of this simulation, a simple network was created with the following parameters:

- 300 Stations
- Uniform random station distribution
- R^2 Path loss
- Minimum transmitted energy routing
- Random traffic model

For the NSMA based protocols, a level of -6 dB was chosen as the noise threshold. Below that level, a station would hold off on transmitting.

Results

Figure 1 shows the SNR results for each of the MACs. ALOHA, as expected, had the lowest SNR performance of all. NSMA by itself was next. The best performance was with TSSS and NSMA combined, followed by TSSS by itself, and then NSMA with backoff.

NSMA with backoff did improve SNR vs. ALOHA, and was only slightly worse than TSSS in terms of SNR. Its latency properties were also favorable.

The addition of NSMA to TSSS improved the overall SNR distribution, and, more importantly, also nearly completely eliminated packets with extremely low SNR.

From figure 2, overall performance can be seen relative to TSSS. Improvements or reductions in SNR translate directly to bitrate.[1] Latency, in this model is inversely proportional to bitrate, so a combined performance factor is obtained. ALOHA is actually an improvement over TSSS due to the greatly reduced latency, according to this measure. This may not be completely accurate, as ALOHA has many packets which have very low SNR, necessitating many retransmissions.

NSMA alone actually performs worse than TSSS due to the loss in overall SNR, and large number of collisions. NSMA with backoff and TSSS/NSMA represent significant performance improvements.

Conclusions

Both NSMA with exponential backoff, and the combination of TSSS and NSMA show promise to improve overall system performance. The SNR improvements allow bitrates to be increased. Further investigation on both of these protocols is warranted.

There is still a lot of room for improvement in MACs for this type of network. An interesting experiment would be to attempt to apply the ideas of MACA [Karn91], in conjunction with NSMA and exponential backoff. This could go a long way towards solving the local collision problem encountered when not using TSSS.

References

[Ettus97] Matthew Ettus. "Power Consumption and Latency as a Function of Path Loss and Routing in Multihop SS-CDMA Wireless Networks." Engineering Design Research Center, Carnegie Mellon University, December 1997. M.S. Thesis.

[Ettus98] Matthew Ettus. "System Capacity, Latency, and Power Consumption in Multihop-routed SS-CDMA Wireless Networks." *Proceedings, 1998 IEEE Radio and Wireless Conference*, Colorado Springs, CO, August 1998, pp 55-58.

[Karn90] Phil Karn. "MACA - A New Channel Access Method for Packet Radio." *Proceedings of the 9th ARRL Computer Networking Conference*, London, Ontario, Canada, 1990.

[OO98] Peter E. Omiyi and Dr. Timothy O'Farrell. "A New Throughput Analysis of a Novel MAC Protocol For CDMA-Based Wireless LANs." *Proceedings, 1998 IEEE Radio and Wireless Conference*, Colorado Springs, CO, August 1998, pp 23-26.

[Shep95] Timothy Shepard. "Decentralized Channel Management in Scalable Multihop Spread Spectrum Packet Radio Networks." Technical Report MIT/LCS/TR-670, Massachusetts Institute of Technology Laboratory for Computer Science, July 1995. Ph. D. Thesis.

---

[1] SNR numbers represent the relative positions of the tail of the SNR distribution. The tail, and not the peak, is important. We don't want average packets to get through. We need nearly all to get through for good performance.
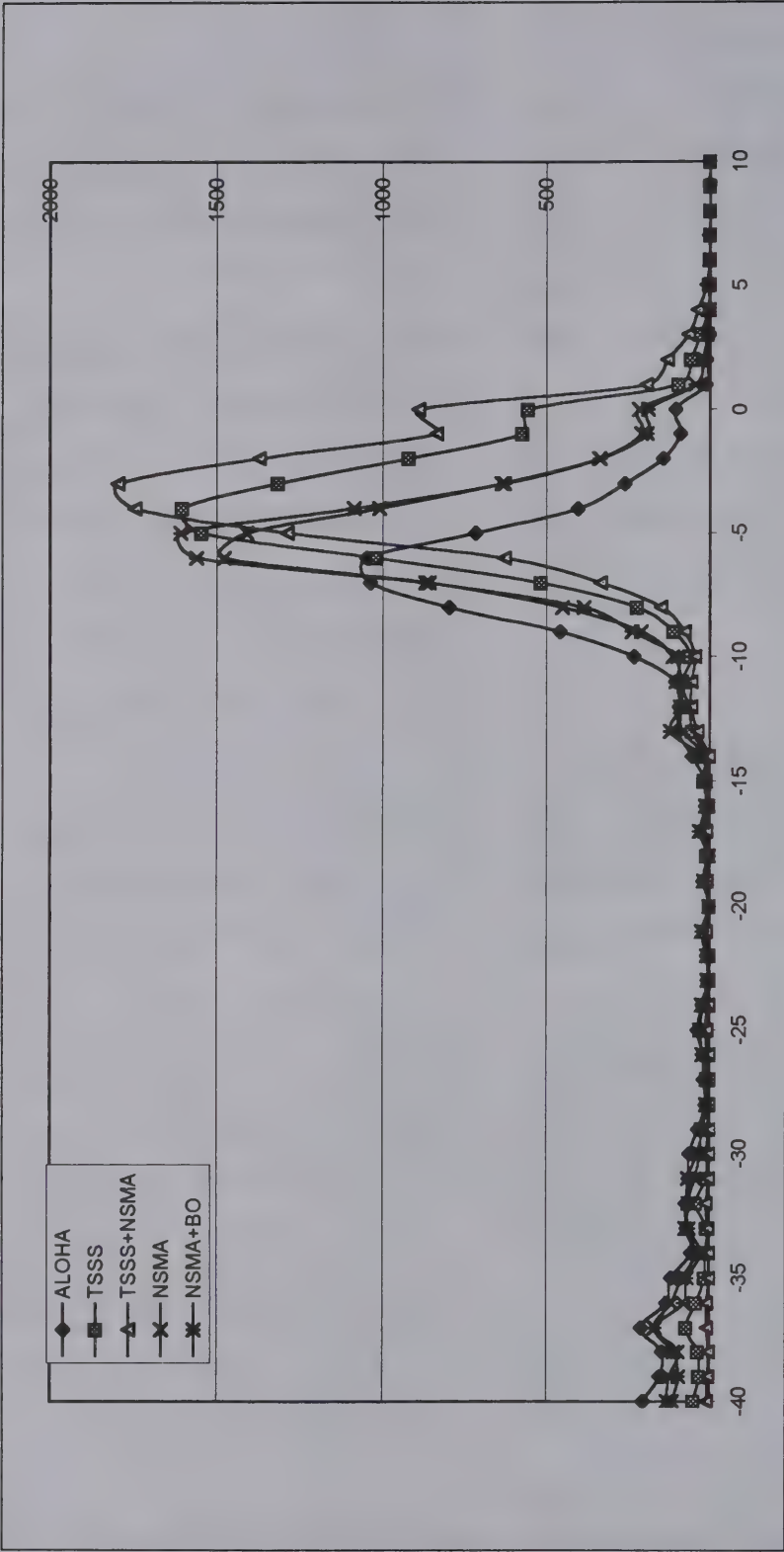
Figure 1

Protocol	Median Latency	Relative SNR	Performance Factor
ALOHA	47	-3dB	1.06
TSSS	100	0 dB	1
TSSS+NSMA	120	1.5 dB	1.17
NSMA	69	-2dB	0.91
NSMA+BO	70	-1dB	1.13

Figure 2

22

An Ethernet Interface for High Speed Packet Radio

Satoshi Funada, 7M3LCG
Packet Radio Users Group of Japan
P.O. Box 66, Tamagawa, Setagaya-ku, Tokyo 158-0094 JAPAN
sfunada@cs.titech.ac.jp

Abstract

In PRUG96 project, we developed an Ethernet interface for high-speed digital transceiver, called IPSM-ZZ. IPSM deals with a only Media Access Control layer protocol. With its partner, Protocol Server deals with upper layer protocols. It allows us to develop upper layer protocols flexibly in common operating systems.

Keywords: IPSM, PRUG96, High-Speed Packet Radio, Ethernet interface

1. PRUG96 system and IPSM

The PRUG96 project, consists of PRUG members, aims to realizing the practical High-Speed Digital Network. The area, we develop, lies from the physical layer to the network protocols.

Figure 1 shows the schematic image at which we set our goal. A small transceiver and a small computer are settled in a lunch-box-size box. We mount this box on the roof, just below the antenna, hence coaxial cable should be short as possible; we use GHz order microwave in our system.

On the other hand, we aim at Mbps order data transfer rate, which is too fast to catch up with for conventional RS-232C I/F. We, the PRUG96 project, decided to use Ethernet I/F, which is popular in
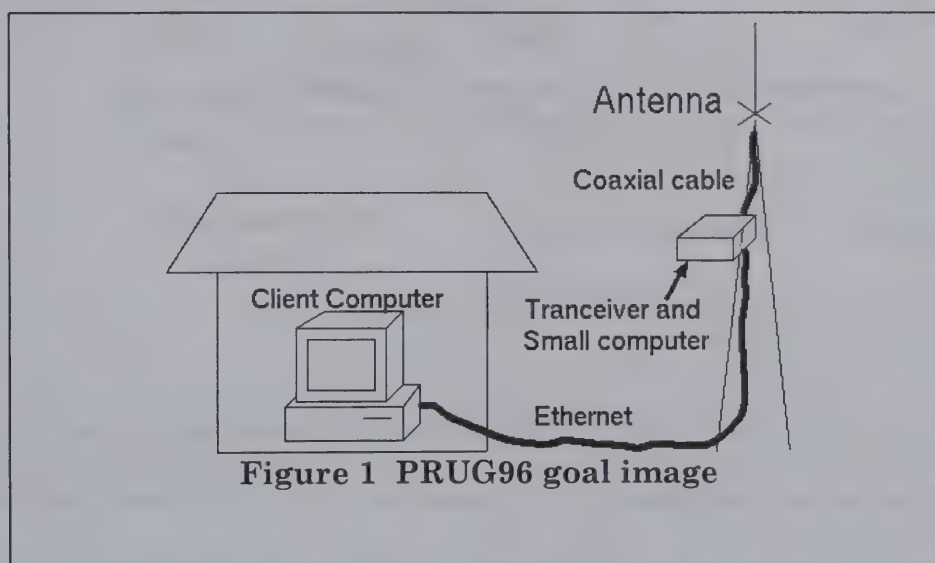
Figure 1 PRUG96 goal image

PCs these days and have the enough ability of 10Mbps.

However, it was very difficult to design physical layer to network layer and to implement them into such a small size device shown in figure 1. In addition, we thought that our system should have a flexibility to make the best use of the resource of amateur radio, such as radio spectra, output power and antennas.

We decided to divide the function of the box into two parts. One part, packet assembling and routing,

upper region above link layer are managed by a PC. Another part remains in the box.

This division makes it possible not only to reduce the cost of the microprocessor in the box, but also to examine and to estimate the routing performance easily, which should be examined again and again.

We designated this method "Protocol Server Method." At first, The data streams to be transmitted processed in the PC called Protocol Server(describe as PS below). Routings and Packet-assembling are done in the Protocol Server and then handled over to the transceiver via the Ethernet, and finally packets are emitted. When the transceiver receives packets, completely reversed procedure is done.

The functions of the box are: To transmit/receive packets; To control the transceiver; To communicate with the Protocol Server(describe as PS below) via the Ethernet. The box contains two pieces; one is the transceiver and another is the small computer, the IPSM (Internet Protocol Shield Machine) which will be described later.

2. PRUG96 mechanism

Before explaining how does the IPSM work, let me talk about the mechanism --- how to transmit/receive a packet in our PRUG96 system.

The PS and the IPSM are connected by the Ethernet. The computer to be accessed with the system, the client, is connected with the PS by the Ethernet or other ways.(See figure 2) Only IPv4 is supported currently. Of course, the client can be connected to the same Ethernet, where the PS and the IPSM are connected. The PS can be regarded as a ' IP router ' from the viewpoint of the client.

Think, the client wants to emit data streams. The client sends a IP packet to the PS. The PS receives a packet and: Adds callsign, comparing inside routing table; Adds Forward Error Collection codes;

Figure 2 PRUG96 system
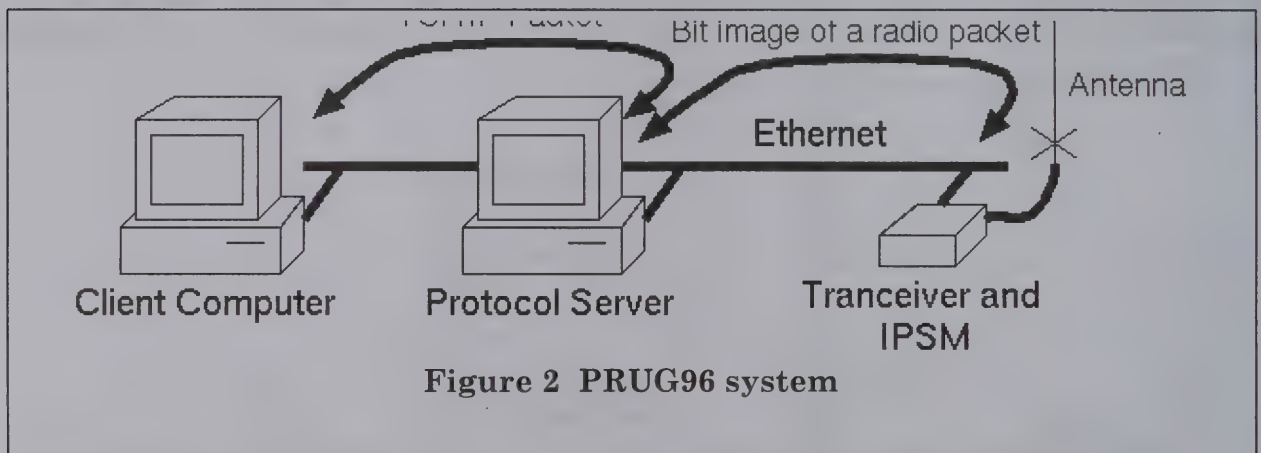
Creates bit images of the packet to be emitted. Then, the PS encapsulates the bit images in a UDP packet and sends to the IPSM. After the IPSM receives a UDP packet, the IPSM removes IP/UDP header from the UDP packet and hands them to the transceiver. Finally, the transceiver emits the packets on the air.

When the transceiver receives a packet, completely reversed procedure is done. The IPSM encapsulates a data image of the received packet in a UDP packet and sends it to the PS, without any consideration whether the packet is addressed to its site or not. The PS analyzed the packet, and if the packet is addressed to its site, transmits it to the client, otherwise throw away the packet.

The IPSM has nothing to do with the TX/RX data streams. The IPSM hand the TX packet to the transceiver and the RX packet to the PS. This method makes the IPSM independent from upper

protocols and therefor we can develop the IPSM alone.

3. PRUG96 MAC layer Protocol

Even though the IPSM has nothing to do with any protocols, it must select the real data from the RX packet. In this chapter, the author will explain about MAC(Media Access Control) layer that the IPSM has to treat.

3.1 Frame structure

In the current version of our PRUG96 system, the TX/RX frame has 1408byte(octet) length, as shown in figure 3. The 32bit length PN(Pseudo Noise) code is added in front of the 1404byte length data stream, in order to synchronize. The reason why the length of a packet is fixed is that we want make it easier to treat the data streams.

If the frame length is flexible, the code which indicates the end of the data   and   data length are required. The IPSM needs to work considering those code and length. On the other hand, if the frame length is fixed, the IPSM is only required to get the 1404 byte length streams after the synchronization. The IPSM has nothing to do with the data stream in a frame. That is the role of the PS.

3.2 Multiple Access
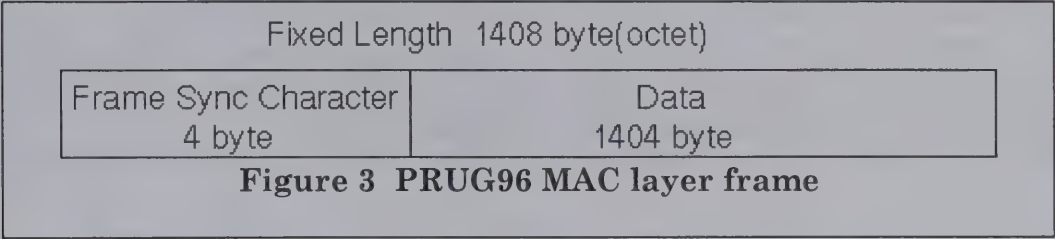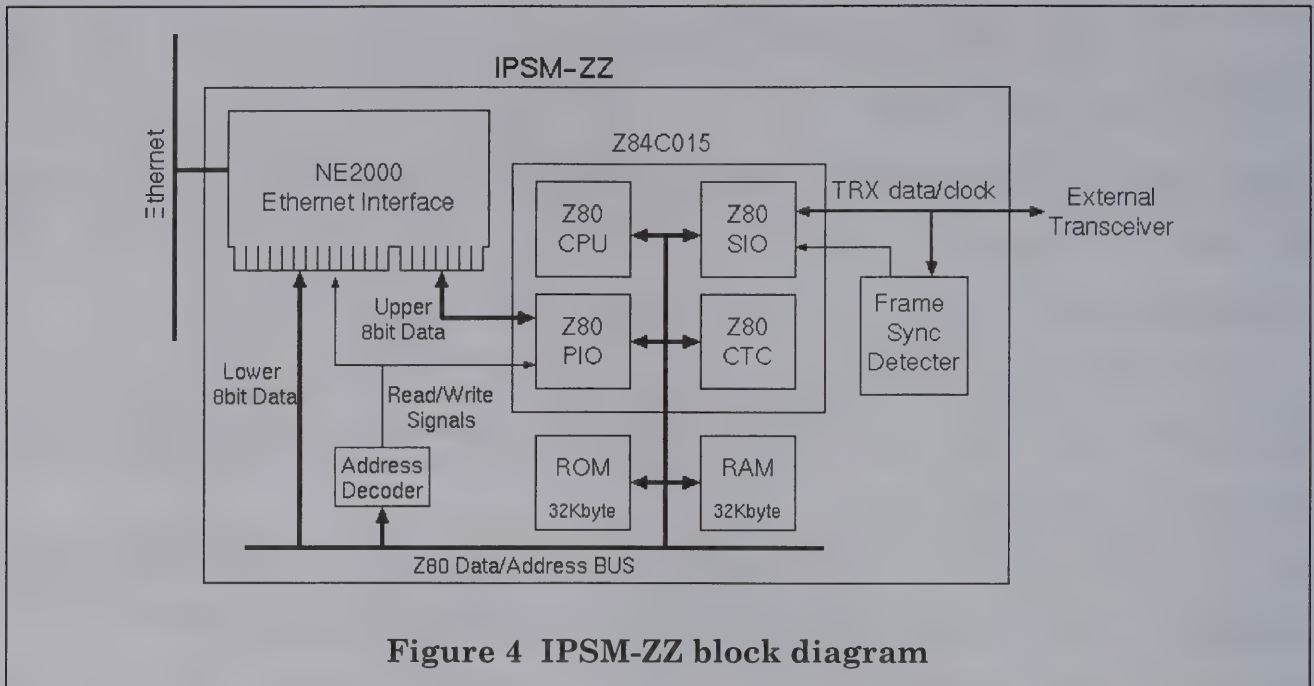
We use CSMA(Carrier Sense Multiple Access) method.

Fixed Length   1408 byte(octet)

Frame Sync Character 4 byte	Data 1404 byte

Figure 3 PRUG96 MAC layer frame

Figure 4 IPSM-ZZ block diagram

4. Hardware Formation

The IPSM-ZZ is one of implementation of the IPSM. The IPSM-ZZ consist of the parts, those are easy to buy. As shown in figure 4, TMPZ84C015BF-10(describe as Z84C015 below) and NE2000 compatible Ethernet card(describe as NE2000 below) are used as the CPU and the Network I/F, respectively. NE2000s are widely used in PC/AT compatibles and the price is reasonable. In addition, NE2000s are easy to install because a NE2000 use only IO ports to communicate with a CPU, where other network cards use both IO ports and shared memories.

Moreover, using table-comparing technique with ROM makes the frame synchronization circuit simple.

4.1 Ethernet interface

An NE2000, 16bit data bus, is connected to 8bit Z80_data_bus via data with converter. A PIO in Z84C015 set to mode3, is used for this converting function, and control signals ASTB and BSTB are connected to NE2000 IO_READ and IO_WRITE inputs. (See figure 5)

The idea is that processing 8bit data from Z80 is latched on PIO port A and it will be passed to the NE2000 at the time following 8bit data is come out from Z80 to be written in NE2000 data port.

The lower 5bit of address signal on NE2000 is directly connected to CPU address bus. Upper 5bit is connected with PIO port B to search and find preset NE2000 I/O address.

Using built-in peripherals reduces circuit complex.

Figure 5 Bus width converter
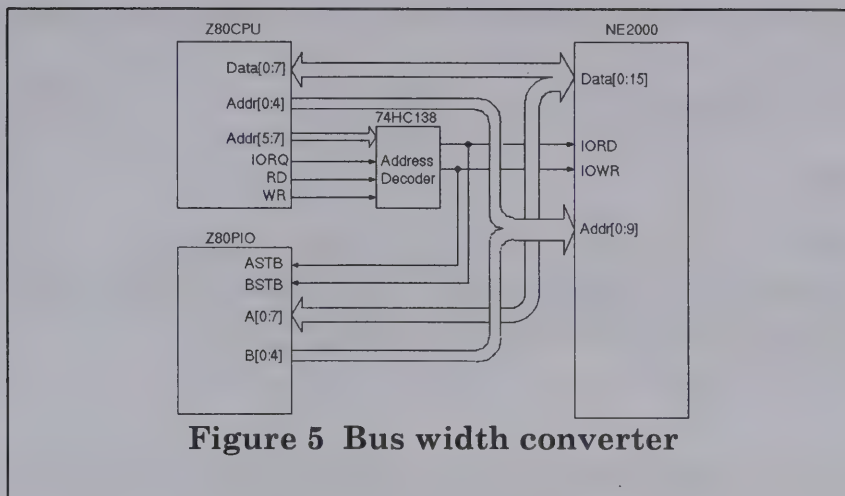
4.2 Frame synchronization block

Figure 3 shows frame structure on the air interface. It consists of 32bit(4byte) frame synchronization header and 1404byte fixed length data. Data block includes FEC data, PS header block, and the upper layer data. The synchronization bit pattern is 0x08f1f059. This synchronization pattern will be used to find start of a frame.

In order to let FEC(Forward Error Collection) function perform correction on occasion of single bit error of twelve bit in data block, synchronization pattern matching block should work on exist of 3bit error in 32bit synchronization pattern. A volume of electronics circuit realizes these functions using logic gates will be larger.

To avoid this matter, EPROM based table searching circuit is selected. (See Figure 6) Received 32bit serial data is converted to parallel data using shift registers to address a cell data of the table in EPROMs. Data shows result of comparison the received 32bit data detected as intended sync pattern or not.

Figure 6 Frame synchronization block
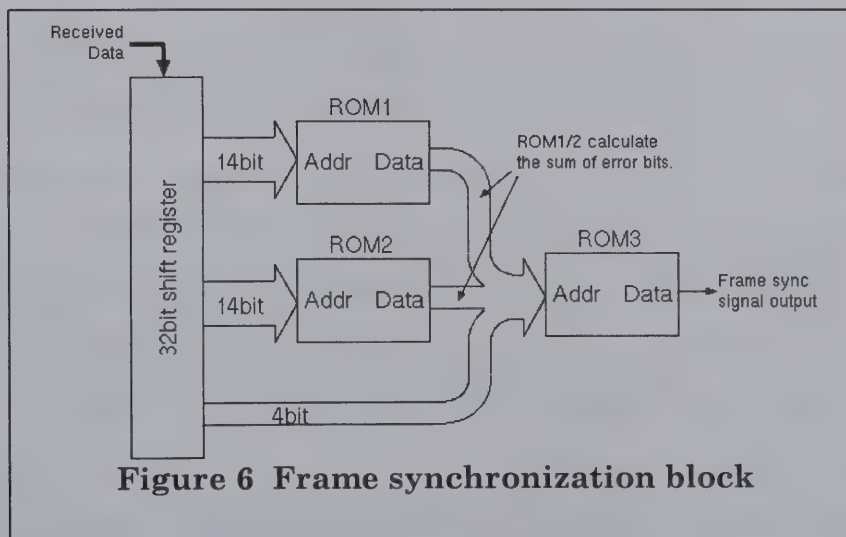
5. Performance

The performances of the IPSM-ZZ are as follows.

(1) Maximum radio speed

Even though the Z80SIO has the ability of 2Mbps, the software can't catch up with it. So that the upper limit of serial transfer speed is about 800kbps. This is due to the defect of RX data just after synchronization. When transmitting, it seems there is no problem.

(2) Delay

It takes about 15ms, to emit a packet since the IPSM-ZZ received a packet from the Ethernet. This is the time to transfer data from the buffer of the NE2000 to the SRAM of the Z80. It's impossible to reduce the time at this moment.

Using 8bit mode of the NE2000s can reduce the delay to almost 2ms. (Implementation is under going, however, not all of NE2000s support 8bit mode.)

(3) Throughput

The transceiver used, when the measurement was done, was the SS data transceiver, produced by root inc., with 808kbps mode. The throughput of the whole PRUG96 system, including the PS and the IPSM-ZZ, using FTP command is as follows: 130kbps(maximum); 80kbps(average).

6. Further work

The current IPSM-ZZ can't make the best use of the transceiver because of insufficient CPU speed. The new model of IPSM is under developing now, aiming to add some functions of the PS into the box.

PIC-et Radio: How to Send AX.25 UI Frames Using Inexpensive PIC Microprocessors

by John Hansen, W2FS
State University of New York
49 Maple Avenue
Fredonia, NY 14063
hansen@fredonia.edu

Abstract: This paper provides step by step documentation of how to implement AX.25 UI frames using inexpensive PIC microcontrollers. It is designed primarily for those who wish to implement packet radio UI beacons for point to multipoint communications. The article assumes some knowledge of programming concepts and PIC microprocessors. It also discusses the limitations that must be overcome in order to build a completely PIC based terminal node controller.

Keywords: AX.25, UI Frames, PIC Microprocessors

Introduction

Over the past several years there has been a growing number of applications that utilize packet radio communication via UI (unconnected) frames. These applications include the PACSAT broadcast protocol, APRS transmissions, and my own HamWeb file transfer protocol. All of these share the advantage that they provide communication of data from one source to many recipients simultaneously.

It is my view that we have only begun to exploit the possibilities of broadcast data in amateur radio applications.   Furthermore, the efficiency of sending data from one source to many recipients is such that it can often overcome the limitations imposed by the relatively low data rates common in amateur radio applications. In short, broadcast protocols can breathe new life into otherwise "slow" data streams of 1200 baud or less. I believe the day will soon arrive that at 1200 baud, UI frames will be the most common means of transmitting amateur packet data.

Currently the principal means of communicating UI frames is to use a standard packet radio terminal node controller. While the price of these had fallen to the $100 range, by using small inexpensive microprocessors for these applications, the cost can be reduced even further.

This paper provides the basic information that is needed to construct AX.25 UI frames using inexpensive, easy to program Peripheral Interface Controller (PIC) chips produced by Microchip Inc. This paper assumes that the reader has some familiarity with PIC chips, how they work and how to program them.   For more information on these subjects, see my recent article in QST.[1]   PIC microcontrollers are available in single quantities for around $6. In larger quantities they are even less expensive. Using PIC microcontrollers, it is possible to build a device that can take serial data as an input and send it out as AX.25 UI frames using a PIC microcontroller for less than $20.

This paper is intended to be generic in nature. While a large variety of PIC microcontrollers are available, this article does not refer to any one particularly, instead the concepts described here apply to at least the entire series of Microchip midrange (series 16) controllers. Further, I did not want to limit this discussion to any one particular programming language. All of the examples presented in this article are in the C language. I have used this language because I think that the code is much

easier to read and understand than is assembly language (the most common means of programming PIC chips). But the basic constructions seen here can be applied in any language (as long as there is an appropriate PIC compiler available).

A Sample Data Stream

For the purpose of this discussion, assume that you want to broadcast the following simple packet:

W2FS -4> CQ, RELAY: Test

This is a simple frame where the source is W2FS-4, the destination is "CQ", there is one digipeter called "RELAY" and the text to be transmitted is "Test". I selected this frame because it is a fairly simple packet, but includes most of the features that you would want to incorporate in a more complex UI frame. In a real world situation, the source of this data could be almost anything... a GPS data stream, an automated weather station, a DX Cluster, a highway traffic information server or perhaps even a generic file server like HamWeb.

What you see above is the packet as it would appear on your TNC. There is actually somewhat more to it than this. The AX.25 protocol divides the packet into seven sections as follows:

1. Flag (s)
2. Address
3. Control
4. Protocol Identifier (PID)
5. Information
6. Frame Check Sequence
7. Flag (s)

Most of the information below describing these fields is a summary of what appears in the AX.25 protocol description.[2] I have simplified this material to only describe what you need to know to send UI frames.

Flags

The flags are simply the hex value 7E (01111110 in binary) sent over and over again when no information is being transmitted. For example, when you set the TXdelay on your TNC to some value, it sends flags (7E's) over and over again for that period. These flags provide the receiver with a clear indication of when one packet has ended and the next is beginning. Thus, there must be at least one flag between two adjacent packets.

Address

The address field contains the destination (CQ, in this case) the source (W2FS-4 in this case), and up to eight digipeters (only RELAY, in this case). Each of "callsigns" in the address field must contain

exactly 7 characters, six for the callsign and 1 for the SSID. If a callsign is less than 6 characters long, it must be padded with blanks. In addition, the receiving station must have some way to determine when the address field has ended (since it could have anywhere from 0 to 8 digipeters in it). This is handled by shifting each of the bits one position to the left so that a 0 appears as the least significant bit. This bit is then set to a one for the SSID (seventh byte) or the last callsign in the address field. For example, the destination callsign would be encoded as follows:

Character	HEX Value from ASCII Table		Shifted Hex Value	
C	43	(01000011)	86	(10000110)
Q	51	(01010001)	A2	(10100010)
Space	20	(00100000)	40	(01000000)
Space	20	(00100000)	40	(01000000)
Space	20	(00100000)	40	(01000000)
Space	20	(00100000)	40	(01000000)

The seventh byte, the SSID, is a bit more tricky. Use the following bit pattern: 011SSSSx where SSSS is the SSID (in binary) and x is a 0 if this is not the last callsign in the address field and a 1 if it is the last callsign in the address field. Since the destination address in this case has no SSID (that is, the SSID =0) and it is not the last callsign, the value should be: 01100000 or 60 in hex.

The next callsign is the source callsign, which in this case is W2FS-4. Using the rules established above we get the following string of bytes for this callsign:

W	2	F	S	Space	Space	SSID = 4
AE	64	8C	A6	40	40	68 (01101000)

There is only one digipeter so it is necessary to set the least significant bit of the SSID of that callsign to 1:

R	E	L	A	Y	Space	SSID = 0
A4	8A	98	82	B2	40	61 (01100001)

Since UI frames are generally broadcast and not directed at any station in particular, the destination space is often wasted with something fairly meaningless (like CQ or APRS). The APRS MIC-E protocol capitalized on this by actually encoding useful position information within the destination address.

Control and PID

Since we are only doing UI frames, the control and PID bytes are pretty simple. Always use the hex value 3F for the control byte and the value F0 for the PID byte.

Information

In this case the information being conveyed is simply the word "Test". This consists of 4 hex bytes as follows:

<div align="center">54 65 73 74</div>

The only thing to be careful about here is that these values are NOT shifted to the left as the address bytes are.

Frame Check Sequence

I was rather disappointed with the description of the Frame Check Sequence that is contained in the otherwise excellent AX.25 Protocol definition published by the ARRL. Here is what it says:

The frame-check sequence (FCS) is a sixteen bit number calculated by both the sender and receiver of a frame. It is used to insure that the frame was not corrupted by the medium used to get the frame from the sender to the receiver. It shall be calculated in accordance with ISO 3309 (HDLC) Recommendations.[3]

Fortunately there is considerable information on the Internet concerning the calculation of this value, which is more often referred to as a "CRC" than an FCS. In addition there is an excellent article in the September, 1986 issue of Byte Magazine on this subject.[4] Rather than review the theory of doing CRC calculations, I will provide a relatively simple mechanism for calculating this value, implemented in assembler and C. This code is included below.

Putting it All Together

Aside from the flags and the FCS (which will be calculated as we go along), our test packet can now be implemented as an array of 27 hex bytes as follows:

C	Q	Sp	Sp	Sp	Sp	SSID=0	W	2	F	S	Sp	Sp	SSID=4
86	A2	40	40	40	40	60	AE	64	8C	A6	40	40	68

R	E	L	A	Y	Sp	SSID=0	Control	PID	T	e	s	t
A4	8A	98	82	B2	40	61	3F	F0	54	65	73	74

Or, as an initialized C array:

```
byte SendData[27] = {0x86,  0xA2, 0x40, 0x40, 0x40, 0x40, 0x60, 0xAE, 0x64, 0x8C, 0xA6,
                      0x40,  0x40,  0x68, 0xA4, 0x8A, 0x98, 0x82,  0xB2, 0x40,  0x61, 0x3F,
                      0xF0,  0x54,  0x65, 0x73, 0x74}
```

Doing it with a PIC Microcontroller

So aside from putting a few flags on the ends and adding the FCS, one simply needs to transmit this set of bytes in order to transmit the UI frame. There are a number of ways that this can be done. First, it is possible to get the PIC itself to actually send the data by simulating a sine wave with something called a resistor ladder. The theory is as follows. You take four (or more) output pins from the PIC and connect four (or more) different value resistors to them. You connect the other ends of these resistors together. Then you step through the output pins to get a rising and falling voltage that simulates a sine wave. This method is documented in a Microchip application note.[5]

A second method is to feed a data stream out of the PIC to a modem chip and have the modem chip generate the tones. There are a number of tone producing chips that will work for this application. Traditionally, packet TNCs have used the Texas Instruments TCM3105. Because this chip is no longer being produced and is getting harder to find and more expensive, I didn't want to start down this road. Instead, I've been experimenting with the MX-COM MX614 chip, which cost about five dollars. It works quite well for this application. It appears that the Philips PCD3311/ PCD3312 chips will also work in this application and cost around $1.50, but I've not had occasion to use them yet.

Even using a modem chip to generate the tones, you can't simply send serial data out a PIC pin and expect it generate packets that can be decoded by other TNCs. There are three reasons for this:

1. Serial data contains start and stop bits. These are not used in packet radio transmissions.

2. The AX.25 protocol specifies that if five consecutive 1's are received in a row, except in a flag byte, a zero should be added after the string of five ones. This is referred to as "bit-stuffing". If you are constructing the AX.25 frames yourself, you are responsible for bit stuffing.

3. Packet radio uses a modulation scheme called NRZI (Non-Return to Zero, Inverted). This means that the ones and zeros are not represented by high and low states (or tones). Rather, a zero is represented by a change in tone (if it was high, it goes low, if it was low, it goes high) while a one is represented by no change in tone. Together with bit-stuffing, this ensures that there will be a tone change at least every five bits, if not more often (except for flags). This helps the transmit and receive timing stay in sync.[6]

In order to implement this in a PIC microcontroller, therefore, you must take the incoming datastream, calculate the FCS, add the flags and route the stream of data to a subroutine that handles the transmission of the data taking into account the proper timing of the bits, bit stuffing, and NRZI. It's a tall order, but it can be easily handled by a $6 PIC chip!

On to the Code

Here is some stripped down C code that will send our sample array as an AX.25 UI frame. This code is written specifically for use with the PIC C compiler made by CCS, Inc. It is the least expensive C compiler currently available ($99). As noted above, the same logic flow could be applied to sending UI frames using assembler. An excellent assembler (MPASM) is available from Microchip, Inc. free of charge. Starting with an overview, the following function will send the packet that is contained in the array SendData.

```
void SendPacket(void){

    fcslo=fcshi=0xFF;      //The 2 FCS Bytes are initialized to FF
    stuff = 0;             //The variable stuff counts the number of 1's in a row. When it gets to 5
                           // it is time to stuff a 0.

    output_high(PTT);      //Turns on the microcontroller Pin that controlls the PTT line.

    flag = TRUE;           //The variable flag is true if you are transmitted flags (7E's) false otherwise.
    fcsflag = FALSE;       //The variable fcsflag is true if you are transmitting FCS bytes, false otherwise.

    for (i=0;i<20;i++) (SendByte(0x7E));       //Sends flag bytes.  Adjust length for txdelay
                                               //each flag takes approx 6.7 ms
    flag = FALSE;                              //done sending flags
    for(i=0;i<27;i++) (SendByte(SendData[i])); //send the packet bytes

    fcsflag = TRUE;                 //about to send the FCS bytes
    fcslo =fcslo^0xff;              //must XOR them with FF before sending
    fcshi = fcshi^0xff;
    SendByte(fcslo);                //send the low byte of fcs
    SendByte(fcshi);                //send the high byte of fcs
    fcsflag = FALSE                 //done sending FCS
    flag = TRUE;                    //about to send flags
    SendByte(0x7e);                 // Send a flag to end packet
    output_low(PTT);                //unkey PTT
}
```

At the heart of the SendPacket function is the SendByte function which is called to send each of the 27 bytes in the SendData array. Here is the SendByte function:

```
void SendByte (byte inbyte){
  int k, bt;
  for (k=0;k<8;k++){                                   //do the following for each of the 8 bits in the byte
    bt = inbyte & 0x01;                                //strip off the rightmost bit of the byte to be sent (inbyte)
    if ((fcsflag == FALSE) & (flag == false)) (fcsbit(bt));   //do FCS calc, but only if this
                                                       //is not a flag or fcs byte
    if (bt == 0) (flipout());                          // if this bit is a zero, flip the output state
      else {                                           //otherwise if it is a 1, do the following:
         stuff++;                                      //increment the count of consecutive 1's
         if ((flag == FALSE) & (stuff == 5)){          //stuff an extra 0, if 5 1's in a row
               delay_us(850);                          //introduces a delay that creates 1200 baud
               flipout();                              //flip the output state to stuff a 0
         }//end of if
      }//end of else
```

```
    inbyte = inbyte<<1;                          //go to the next bit in the byte
    delay_us(850);                               //introduces a delay that creates 1200 baud
  }//end of for
}/end of SendByte
```

Note that for each byte, the data is transmitted least significant bit first (that is from right to left, rather than from left to right). The function delay_us is a routine shipped with the CCS C compiler. It is supposed to create a delay of 850 microseconds. You might think that this is too long a period a time since 1200 baud would normally require that each bit last exactly 833 milliseconds   (1 sec/1200). The CCS timing routine is not exactly accurate, however. Experimentation revealed that a value for the CCS function of 850 resulting in timing that is correct for 1200 baud.

The flipout function changes the state on the output pin when a zero is being sent. It is as follows:

```
void flipout(){                                  //flips the state of output pin a_1
        stuff = 0;                               //since this is a 0, reset the stuff counter
        if (!bit_test(port_a,1)) (output_high(pin_a1));   //if the state of the pin was low, make it high.
                else (output_low(pin_a1));       //if the state of the pin was high make it low
}
```

Finally, we need the routine that actually calculates the FCS. The FCS consists of two bytes, which I have called fcslo and fcshi.   These are both initially set to FF. In this example the FCS will be calculated on a bit by bit basis. Algorithms exist that can calculate the FCS either a bit at a time or a byte at a time. Here is the calculation routine:

```
void crcbit(byte tbyte){
#asm
   BCF    03,0
   RRF    fcshi,F                  // rotates the entire 16 bits
   RRF    fcslo,F                  // to the right
#endasm
   if (((status & 0x01)^(tbyte)) ==0x01){
       fcshi = fcshi^0x84;
       fcslo = fcslo^0x08;
     }
 }
```

The function parameter tbyte is either the byte 0000 or the byte 0001 corresponding to the value of the bit that is currently being transmitted. I have used three assembly language instructions in the beginning of this function (between #asm and #endasm) because there is no simple means of rotating a 16 bit value in the CCS C implementation. These three assembly language instructions simply move the 16 bit value one place to the right, with the previous least significant bit being placed in bit 0 of the status register in the PIC chip. The next line (the line that begins with if) performs an exclusive or (XOR) on this bit from the status register and the bit that is being transmitted (tbyte). If the result is equal to 1, the FCS is XOR-ed with the hex value 8408. If the result is equal to 0, this

latter step is not performed. Either way, the new value of the FCS is preserved in fcshi and fcslo. This procedure may seem rather arcane, but it does work. For a discussion of the theoretical reasons behind this procedure see the 1986 Byte Magazine article.

A PIC-based TNC?

From the above discussion it is clear that it is not all that difficult to send AX.25 frames using a PIC chip. There are a wide range of beacon type applications where such a device could be very useful. To go one step further, does this mean we could build a PIC-based TNC for very little money? Unfortunately this is not a trivial matter. My near term goal is to find a way to build a 1200 baud transmit module that will take a continuous data stream, convert it into packets and transmit it. A similar module on the other end of the link would undo the process. Using this mechanism you could transmit virtually any serial data stream from one point to many points using existing amateur radio transceivers without conventional TNCs. My intermediate term goal is to build a full duplex stand-alone 1200 baud (and then 9600 baud) KISS TNC using these inexpensive chips. If this could be accomplished it would have a myriad of applications including very inexpensive 9600 baud amateur satellite modems.

There are a number of hurdles to be overcome before any of this is possible. Starting on the transmit side, the basic problem is that the device must receive data via a serial link at the same time that it is transmitting data over the radio. The beacon style device discussed in this article takes existing data that it obtained from whatever source and transmits it using AX.25. For the purposes of this device, it is assumed that the data stream is not continuous. If the input data stream is continuous, however, while it is transmitting the first packet, it must also be accumulating data for the second packet over the serial link. Some buffering would also be required since there is not a bit for bit correspondence between the serial data stream (which includes start and stop bits) and the radio data stream (which includes no start and stop bits, but does include addresses, PID and control bytes, the FCS, etc.).

The receive side may actually be a bit more difficult. This is because the incoming packet must be received in its entirety in order to calculate the FCS before it is forwarded out the serial port since packets with incorrect FCSs should be discarded. Thus there must be enough buffer space to hold the entire packet. Most packet applications, including all of the amateur digital satellites, are limited to PACLENs of 255 characters. There are PIC microprocessors with enough on board memory to handle this available in the $10 range. However, some protocols are now using packet lengths in excess of 1K. No PIC contains this much on board storage, so some external SRAM would be required. This, in turn would involve using a PIC with a substantial number of I/O pins (for both the data and address lines).

Conclusion

Contrary to popular opinion, the most significant limitations in packet radio today are not technological. Amateur radio operators are only beginning to scratch the surface of the range of things that can be accomplished with the technology that is already available. In addition to the quest for faster speeds, we should also focus on new applications that can be developed with the slower

speed digital technologies that can piggyback on conventional FM radio channels. One key to doing this is to develop extremely inexpensive packet radio interfaces. PIC chips can provide a means of doing this.

[1] Hansen, John A., " Using PIC Microcontrollers in Amateur Radio Projects" (*QST*, October, 1998).

[2] Fox, Terry L. *AX.25 Amateur Packet-Radio Link-Layer Protocol, Version 2.0* (Newington, Ct: ARRL, 1984).

[3] *ibid.*, p.4.

[4] Morse, Greg "Calculating CRCs by Bits and Bytes" (*Byte*, Sept 1986, pp. 115-124).

[5] Microchip Application Note 655, "D/A Conversion Using PWM and R-2R Ladders
   to Generate Sine and DTMF Waveforms" .
   http://www.microchip.com/Download/Appnote/Category/16CXX/00655a.pdf

[6] For a more complete description see: McDermott, Tom, *Wireless Digital Communications: Design and Theory.* (Tucson:
   TAPR, 1966) pp. 121-126.

A New Vision for the Amateur Radio Service

Dewayne Hendricks, WA8DZP
Greg Jones, WD5IVD

Vision Statement Concerning the Future of Amateur Radio

Amateur radio as a hobby has reached an important turning point. Many can point to various examples of why things are changing; however, some of these examples are real and some are only periodic in nature, but the trend of activity and interest now as compared to five or even ten years ago is changing. The real issue which we must face is 'does the amateur radio service (ARS) base its future on the precepts created and tested over the last twenty years or do we look at new and novel ways of growing, sustaining, and protecting the hobby that we love?'

As active members in the ARRL, since first licensed, active members at various internal levels of the League, and very active in the area of amateur radio technology advancement that TAPR represents, we would like to take a few moments of your time to share some important thoughts on the matter.

The Commercial Future of Amateur Radio and how the ARS can benefit from the change

Amateur radio has prospered over the last twenty years as commercial manufactures were able to grow radio sales in the US, with the amateur radio community as a secondary market to their already existing commercial markets. This resulted in a tremendous growth and usage of VHF/UHF and to some extent, HF, over the last several decades.

We now find many amateur radio vendors and manufactures reducing their presence or even leaving the amateur radio market for other markets or to refocus on their older commercial markets as new communication systems threaten to take market share away. Some stores that have been in existence for sometime have even begun closing their doors. This is to be expected with the sales of amateur radio equipment dropping off. Keep in mind that some say this is sunspot related, but can sunspot activity also explain the drop in the VHF/UHF market as well? Amateur radio is in the midst of a paradigm shift from the vast majority of communicators currently on the bands to a more balanced population representing technical, experimental, and hobbyist who just like to communicate with radios.

As vendors continue to leave the amateur radio market, it is up to organizations like ARRL, TAPR, and AMSAT (the three major non-profit amateur radio organizations in existence today) to grow our technology internally, instead of waiting for external forces to discover amateur radio as a market. If we wait for external market forces to come into play, we will find that these companies will probably rather seek out commercial markets where there is more profit potential, then the hobbyists market which uses our radio spectrum for recreation, learning, and public service.

TAPR has begun working in this direction, by working with the remaining manufacturers and looking elsewhere to non-traditional funding

sources like the National Science Foundation (NSF). We see grants and other such efforts as just a beginning in which to grow more money and more research that will hopefully benefit all of amateur radio in the long term. However, the amateur radio rules are going to need to be more proactive to allow for these types of new technology-oriented ventures to take hold and grow. Amateur radio must have rules that allow experimentation with new modes, without the need to get an STA or waiver each and every time someone wants to do something new. If we don't see this necessary flexibility in the future we will find that most potential amateur radio projects will end up operating under Part 5, Part 15, or any of a number of other services. Or worse yet, amateur radio operators will just ignore the current rules and build and operate equipment to provide the kinds of services that they desire.

While amateur radio has a great history with a rich tradition of introducing new ideas and technology, that process seems to have slowed as more communicators joined the hobby. It became more important to make sure these communicators and people who simply enjoy the hobby aspect of the service had no problems operating and the introduction of new systems and experimentation slowed as a result. It is true that while we have seen a lot of work in new digital and RF areas niche interest, none of this research has been widely adopted or been beneficial to the larger majority of the members of the service.

As an example, an organization like the ARRL is in a position to greatly influence the realization of expanded growth of amateur radio by supporting the efforts of small, innovative companies making contributions to the hobby and not large manufacturers whose primary business and marketing interests are in other areas than amateur radio. It is in the best interest of amateur

radio service (ARS) to grow this cottage industry, because these groups could well become the next Collins, Drake, and other amateur radio-founded companies in the future. What we see today is that various members of the service are starting companies, but these new organizations are focused on other services, because the current FCC rules and the 'climate' of the hobby don't really allow for the easy introduction of new types of technology. These same companies are the ones that are now asking for more spectrum from the FCC for their products and services -- and where do they look ? They look to amateur radio spectrum because they understand full well just how under utilized that spectrum really is.

What is to keep the ARRL or TAPR from creating its own "Co-Op" approach like REI or many other such organizations? Together both organizations have the membership base to easily support such an effort and the potential impact on the purchasing power from the total membership could lead to an environment where product development decisions were being made based on the needs of amateur radio operators in the US, instead of those requirement being secondary to existing market needs and requirements as viewed by technology manufacturing companies located in other countries.

Experimental and Technological development are keys to the future

It has been a concern of ours and TAPR's for some time that there is a tendency to resist change when something new or novel appears on the amateur radio scene. TAPR, AMRAD, AMSAT, and other organizations represent the spirit of change and development within the ARS. Amateur radio can either choose to support various efforts within the community for the most advancement of new

technology or wait for external commercial forces to quickly take advantage and look for additional spectrum, most likely being the current ARS allocations. Not many amateur radio groups or individuals can sustain the effort required to make change happen under the current restraints to the introduction of new technologies. The expense of development, manufacturing, marketing, and to some extent the rules themselves affect the introduction of new technologies to the service. Most new operating interests within the hobby have been a result of the usage of other external technologies (i.e. Personal Computers, Internet, etc.), not of something grown from within the hobby itself.

It is important that ARRL,TAPR and AMSAT watch out for the interests of its diverse membership, but at the same time it must be working on providing support for various efforts elsewhere in the community that are emphasizing new technology and change. The ARRL doesn't have to lead, but it must be fully supportive of change and be willing to facilitate it as much as it can. While an open support policy might threaten some, it is imperative that ARS grow from within and it is equally important that the organizations take a leading role in helping to encourage the growth of new operational modes and techniques.

Amateur Radio should develop it own spectrum sharing partners

With regard to spectrum, we believe that the ARS can either continue to defend the spectrum we have, or look for those services whom we want to share our bands. We have to locate others that can help fully utilize our valuable spectrum, but not take away from the mission and operating flexibility of the ARS. This could be the form for instance of the creation of a low-power educational wireless service which could be overlaid on some part of the existing ARS spectrum or some other similar approach. The League successfully used this tactic several years ago when it joined with Apple Computer in lobbying the FCC to designate the 2390-2400 MHz band as a shared band with only the ARS and U-PCS as the incumbents.

The ARS should think about what services would be the most 'tolerable' on our bands. We can't say no to everyone forever, because that will likely result in our losing even more spectrum over time. By finding and locating or creating friendly sharing partners we 1) protect our spectrum on our own terms, 2) create a commercial need for equipment, if done correctly amateurs can leverage these devices into operational 'ham ready' units, and 3) bring users from the shared spectrum services into the ARS where applicable. This is one reason we have suggested the educational communication service concept. It would get members of the ARS into schools helping install wireless networks that might have rules like Part 15, but this direct contact with schools could easily lead to students getting interested in amateur radio because of the close working relationship formed when the local/regional ARS organization helps get the school wireless connections to the Internet.

TAPR Response to ARRL New Repeater Concept

TAPR has been working on a new 'high concept' repeater system that makes use of spread spectrum technology, in particular, frequency hopping to act as a stepping stone to a new generation of devices that can provide new levels of function and operational flexibility to the amateur radio community.

TAPR on its own as been working in this direction for the last two years. Its first steps in this direction was the submission to the

NSF of a proposal for what has come to be called the 'Internet Access Radio' (IAR) in the Fall of 1996. The first member in a family of such radios is currently under development and information on it can be found on the TAPR website at: <http://www.tapr.org/tapr/html/taprfhss.html>.

TAPR believes that today's communications technology is moving toward all digital transmitters and receivers. These advances in technology, combined with the swift evolution of cell based transmission and switching protocols is opening up a new set of possibilities for unique new services utilizing intelligent networks which will contain smart transmitter, receivers and switches. Today's Internet is perhaps the best example of the a self regulating structure which embodies these new technological approaches to communications in the networking domain. However to date, many of these innovations have not made it over to the wireless networking arena. What TAPR feels that the radio networks of the future will involve a mixture of links and switches of different ownership, which terminate at the end-user via relatively short distance links. What will then be required is an built-in, distributed, self-governing set of protocols to cause the networks behavior to make an more efficient use of a limited, common shared resource, radio spectrum. Creating such a self-regulating structure for the optimal sharing of spectrum will require much effort. One of the major problems which stands in the way of these new approaches today is the current FCC regulatory environment and the manner in which spectrum is managed and allocated under its rules.

One of the major hurdles that an wireless entrepreneur faces who wishes to develop innovative new communications products which involves radio is access to the requisite amount of spectrum. This process makes the involvement of the wireless entrepreneur with the government mandatory, which immediately puts them at a disadvantage when compared to entrepreneurs in the computer sector where government involvement is minimal. As a result, innovation has occurred at a much slower pace since the use technologies such as spread spectrum require the use of more spectrum and not less in order for their advantages to become apparent when it is used for high-speed data transmission.

Historically, the current regulatory approach to radio has been based upon the technology that was in use at the time that the Communications Act of 1934 was framed, basically what we would call today, dumb transmitters speaking to dumb receivers. The technology of that time required reserved bandwidths to be set aside for each licensed service so that spectrum would be available when needed. Given this regulatory approach, many new applications cannot be accommodated since there is no available unallocated spectrum to 'park' new services. However, given the new set of tools available to the entrepreneur with the advent of digital technology, what once were dumb transmitters and receivers can now be smart devices which are capable of exercising greater judgment in the effective use and sharing of spectrum. The more flexible the tools that we incorporate in these devices, then the greater number of uses that can be accommodated in a fixed, shared spectrum.

While the IAR proof-of-concept (POC) radio is under development, TAPR intends to make the case to the FCC that the current rules should be changed to reflect that use and advantages that smart spread spectrum packet radio devices can realize. TAPR's position is that a major improvement in spectrum use is feasible in the concepts to be

employed in the IAR POC radio are put into widespread use. However, given the radical nature of some of the approaches in this project, it is appropriate to first, confirm the technical theories that we are putting forth and then to define the operational parameters for the implementation of these theories once they are confirmed. Then we will be able to approach the Commission with proposals that have a sound basis in fact and which should hopefully then be acted upon in a favorable fashion.

While development of the IAR POC is underway, TAPR has several projects underway that utilize existing Part 15 spread spectrum radios that are being adapted to meet amateur radio operational requirements and which will be used for general packet radio and Internet access over wide-areas. One project uses OEM modules from Lucent Technologies and the other uses a radio provided by a member of TAPR's sister organization in Japan, the Packet Radio User's Group (PRUG).

Much of what we have in mind can be accomplished today with existing Part 15 radios. One of the author's of this article has such a system currently up and operational in the San Franciso Bay Area. The system uses two mountain top sites and can currently cover all of the South Bay Area, providing voice and data services to users at ranges up to 20 miles. Here are the characteristics of the system:

- Operates on 2.4 GHz.
- Radios use FHSS half duplex. Output power is 1W. EIRP is within FCC limits of 4 W EIRP.
- TCP/IP protocols are used.
- Accepted Internet protocols are used to handle voice and data traffic.
- System can be accessed by any device that uses the TCP/IP protocols and a similar dataradio.

Here are some of the things that this POC radio system can accomplish:

o   Can handle several separate voice conversations, bulletins, and data streams simultaneously?

Yes, using standard Internet protocols. Uses the H.32x standards.

At the core of the H.323 standard is a method for managing network latency, or the time it takes to send and acknowledge a packet. High-latency networks such as the Internet, where data packets must jump through many routers and subnets, have a tendency to wreak havoc on audio and video synchronization. To address this shortcoming, H.323's Real-Time Transport Protocol (RTP) time-stamps and sequences packets and reduces delays.

H.323 also specifies the coding and decoding of video and audio signals, optimizing data for lower bit rates and low-bandwidth connections. H.323-compliant products are now quite common on the market with Microsoft's NetMeeting being a good example. More information on H.323 can be found at: <http://gw.databeam.com/h323/h323primer.html>.

o   Supports duplex (just like a telephone) and conferencing (just like a teleconference)?

Yes, again using standard Internet protocols, even though the acutal radio link is half duplex.

o   Lets you know who else is monitoring and lets you contact them without interrupting anyone else?

Yes.

o   Is resistant to deliberate interference, and allows the control operator to "lock out" stations that are not following the rules?

Yes. We have full control to lock out users as required by a number of different methods.

o   Can share its operating frequencies with several similar repeaters nearby, with little degradation in the performance of any of them?

Yes. We are able to add new mountain top sites without the need for coordination.

o   Lets you use one radio to access all of these functions, and others such as PacketCluster and APRS, simultaneously?

Yes.

o   Puts the amateur allocations above 1 GHz to more intensive use?

Yes. In this case, 2.4 GHz is used.

So it would seem from TAPR's work and experiences to date that we are really not to far from demonstrating a system to the amateur radio community that is quite similar to that proposed by the League. To get things moving to the next step, TAPR would like to propose the following to the amateur radio community in general:

o   Setup a meeting as soon as possible between TAPR and the other amateur radio organization to discuss this effort in more detail. The end result of that meeting would be a working paper and a set of recommendations to both organizations as to what next steps would have to be taken to make this concept a reality.

o   Install and play with one of these Part 15 systems in different part of the country. Such a system could be procured and deployed for a total cost of less than $10K. TAPR

would be happy to provide all of the necessary specifications.

Conclusion

We believe that amateur radio has been at a crossroads for the last several years and continues to wait for the "light to change" to indicate what the future will really hold in store for the service. The ARRL, TAPR, AMSAT, and other technology-oriented groups must take the initiative and forge ahead into the future on our own. We need to be proactive to change and challenges, and not take a position of "wait and see" for attitudes to change. There will be those members in all of our organizations that will hate what the future will bring, but past history and experience shows us that adopting a position of limited or no change only means that the change and growth will occur elsewhere. Change does not mean the total abandonment of the past traditions that we believe have made the amateur radio service what it is today. We can either bring about increased growth in our ranks or see that growth occur on the Internet and other areas that many of our members will perceive as much more fun and enjoyable ways to spend their time. Not following the course of change might be the wise political approach to adopt for now -- but is it unlikely to be the most productive one.

The issues and actions the we have raised are just some thoughts about where amateur radio is today and where it might be going These are just first steps towards a new future and many more will be required to effect any real change. Long range planning is certainly important, but with the increased pace of change in society and the technology sector, amateur radio needs to take a fresh look at where it has been and just where it would like to go.

APRS QSY from 145.79 to 144.39

Greg Jones, WD5IVD
President, TAPR

Frank Bauer, KA3HDO
Vice President of AMSAT Manned Space Operations

Stan Horzepa, WA1LOU
TAPR APRS SIG Chair

Steve Dimse, K4HG
APRS QSY Proposal Liaison

Abstract

This little adventure began over a year ago when Frank Bauer, KA3HDO, Vice President of AMSAT Manned Space Operations approached the community with the issue of potential interference to the future of amateur radio operations aboard the International Space Station. Much discussion took place before the final proposal was put forth at the 1997 ARRL and TAPR Digital Communications Conference. At the conference, Frank presented his paper "Amateur Radio on Manned Space Vehicles: Improving Amateur Radio's Future Through Enhanced Space Frequencies." [Bauer, 1997] This paper discusses the basic issues of the proposal. What we have a year later is a nearly complete process of a large section of the amateur radio community voluntarily changing frequency. It wasn't easy or without debate, but the process showed that with adequate communications and lots of time discussing and educating people on the reality of the situation, that change can happen in the amateur radio service in a cooperative manner. This paper will discuss history leading up to the QSY, the APRS QSY proposal, major events in the process, results of the APRS QSY survey, current status of the QSY, who received money from the QSY fund, and finally what lessons were learned along the way.

History of 145.79

As was posted in a message by Tom Clark, W3IWI, [Clark, 1997] the reason APRS started on 145.79 dates back to the 1990 era of packet radio in the local Maryland and Washington DC areas. Several hams in the area including Tom Clark, W3IWI, decided to open up the 145.50 - 145.80 segment to packet activity. They felt that the five slots in the 145.01-145.09 segment were inadequate for all the PBBS, linking, DX Cluster, simplex rag chewing, etc that was happening on packet radio. The local packeteers prevailed on the local FM folks to quadruple the available space which the packet community would administer. They elected to follow the local pattern of 20 kHz channelizing, using the odd tens kHz slots (300 kHz of spectrum yielding 15 available channels). Tom at that time, being very involved in AMSAT, understood the band-edge problems with regard to the satellite sub-band and thus placed 145.79 as a "reserved for experimentation." The plan was for this frequency to be used for new technologies, especially modems.

At about this time, Bob Bruninga, WB4APR, began testing a <UI> frame (unconnected datagrams) protocol and asked about getting a frequency to test what he saw as a way for low-powered mobiles to transmit information to nearby stations working like the cellular

telephone network with limited coverage. Tom suggested he use 145.79 with the implied reality that it was adjacent to the satellite band, that it was subject to occasional QRM, and that the concept was experimental.

As we all know, Bob's modest request a few years later became APRS, and 145.79 became APRS's national home.

The crux of our story begins when the post-Challenger shuttle program resumed and AMSAT Manner Space Operations resurrected the idea of SAREX carrying amateur packet hardware (the SAREX TNC/ROBOT). AMSAT tried very hard to find a suitable frequency for the SAREX ROBOT. Since it involved both up- and down-links, and since most radios were built for 600 kHz splits, they tried pairing frequencies like 144.950 with 145.550. This choice was not very well received by some packet radio communities. The reason being that in the late 1980's the 145.01-09 packet frequencies had been added to with the addition of 144.91-99. When SAREX began operations on 144.950, there were a lot of individuals who had packet radio systems running on 144.950 who were very unhappy about the intrusion.

This was one of the first cases of how do you fit into differing world wide bandplans operating frequencies for space missions that do not interfer with anyone else. The real problem here is that in Region 1, the 2M band is only 2 MHz wide (144-146 MHz). The situation is made even more difficult, since the band plan has to be agreed to by 50+ countries. For reasons that have to do with International governmental treaties negotiated at the WARCs, the amateur satellite service is restricted to the universal international parts of the bands, so any spacecraft using 2M must operate between 144 & 146 MHz -- no choice! In that context, the satellite community has convinced the entire, world-wide amateur community (thru the IARU) that 10% of the worldwide 2M band -- 145.80-146.00 -- must be reserved for space activities. Thus SAREX's usage of 145.55 was not well received by the Europeans either!

Now let's add MIR into the mix. The suggestion for the use of 145.20/145.80 for MIR came from an IARU Region 1 conference (Tel Aviv), and after the idea was announced to the rest of the world, it became obvious that it was not a good GLOBAL choice since 145.20 was heavily used elsewhere (in Europe, 145.20/.80 was a repeater pair which was being phased out -- which is why the idea made some sense in Region 1). The problem in the US is that APRS is/was on 145.79. Although the proposal made since in Region 1, it didn't necessarily fit into Region 2.

As Tom points out in his e-mail [Clark, 1997], "The real underlying problem is that the 2M band is crowded, especially in Europe. 2M links for MIR/SAREX/ISS are desirable since EVERYBODY already has the necessary radios. The problems are compounded because the 2M band has, in general, been treated as a local coverage resource with emphasis on terrestrial repeaters -- except for the bottom-end "DX" and the top-end satellite chunks, the people who dole out frequency slots wear a "100 km radius" (i.e. 60 miles) localized set of blinders. The local repeater operator/coordinator has virtually no interest in what happens 1000 km away! Witness the fact that different parts of the USA adhere to 15 & 20 kHz channel spacings as a local option!"

Historically, we find ourselves here in the US using 145.79 because APRS at the time was considered a local experiment when it began. Add to this the reemergence of amateur radio activities aboard manned space missions that have very limited operating frequency constraints and the potential problem of interference between the two groups is very high.

The Proposal

The proposal to QSY APRS presented at the 1997 ARRL and TAPR Digital Communications Conference wasn't the first such suggestion. There had been several discussions and proposals before this one that looked at the issue of APRS being on 145.79 and 145.80 being used for space based operations. What made this proposal different was that all the elements were in place for a successful proposal. There was a clear outstanding need to reduce near band interference before the International Space Station began amateur radio operations. The facts already showed that orbiting crews endured significant frequency interference issues to achieve success that many simply turned off the radio. Thus, these frequency problems have limited the growth and success of this communication medium. The real downside to the interference issue was that the full potential of this facet of amateur radio to infuse new blood into the hobby through educational opportunities for students and its positive experience to the community has been somewhat stunted due to these frequency problems.The potential amateur radio promotion for successful amateur operations on the ISS is not disputed by anyone. How can anyone argue against the fact that communicating with astronauts and cosmonauts is an exciting and challenging facet of amateur radio. The APRS community was operating one two main frequencies. 145.79 and 144.39. The proposal to move everyone to one single frequency to help with creating a true national/international agreement (with Canada) was a seen benefit to the now rapidly growing APRS community that is seeking increased coverage and ease of use between areas of operation. After much education on the subject, most could see the problem with the location of a frequency on board MIR and the ISS due to international limits for frequency selection.

In addition, several new items that past proposals didn't have were added. These included that each of the three major organizations (TAPR, AMSAT, and the ARRL) would donate money towards a QSY fund to help with the relocation. After all the discussion and debate, only $1500 was spent towards helping QSY. Most sites simply changed frequency or paid for the cost locally. All three major groups (TAPR, AMSAT, ARRL) showed support by passing a motion on the issue at their board of directors meeting. A committee was formed to help coordinate the efforts of the QSY and open debate then began. The committee consisted of: Stan Horzepa, WA1LOU, TAPR APRS SIG Chair, Steve Dimse, K4HG, APRS QSY Proposal Liaison, Greg Jones, WD5IVD, President, TAPR, and Frank Bauer, KA3HDO, Vice President of AMSAT Manned Space Operations.

TAPR, AMSAT, ARRL

Once the three major organizations involved passed a motion at their board of directors meeting, the APRS QSY committee felt we had a chance to make this proposal work. Without the support from each of these groups, the proposal would have lost a lot of its positive energy. With the passing of each motion, the proposal gained strength that this was finally the right mix to solve the problem for everyone.

TAPR Board of Directors Positions Statement

1) TAPR, in support of its APRS SIG and the organizations of many APRS users, recognizes that APRS is a vital and exciting facet of amateur radio.

2) TAPR supports the experimentation of APRS through various amateur radio satellites and the International Space Station.

3) TAPR endorses the concept of an APRS-QSY Fund and will help set up and administer such a fund when the time becomes necessary to facilitate the potential QSY of APRS U.S. infrastructure.

4) TAPR approves a donation of $500 to support the QSY initiatives when the fund is established.

AMSAT Board of Directors Position Statement

The AMSAT- also agreed (in cooperation with the Tucson Amateur Packet Radio (TAPR) organization) to help an ongoing effort aimed at minimizing the impact of moving a large number of current Automatic Packet Reporting Systems (APRS) users off of 145.79 MHz. The Board agreed to donate up to $500 to a fund to help defray needed expenses of various fixed frequency APRS node operators in finding another "home" for their APRS operations in the USA. If the shift to another frequency eventually proves acceptable to the APRS community, it would help resolve one of the last remaining issues in clearing 145.80MHz for worldwide use by MIR, SAREX, and ISS.

ARRL Board position statement on QSY [ARRL, 1998]

Whereas, the ARRL recognizes that APRS and SAREX/ARISS are vital and exciting facets of Amateur Radio, and Whereas, the ARRL recognizes the unique needs of APRS and SAREX/ARISS for nationwide frequencies, and Whereas, the ARRL supports the experimentation of APRS through various Amateur Radio satellites and the International Space Station, and Whereas, TAPR and AMSAT-NA have endorsed the APRS/Manned Space alliance and the "APRS QSY Activity" and have each pledged up to $500 to the "APRS QSY Donation Pool," Be it resolved that the ARRL endorses the concept of an APRS/Manned Space compromise as a mechanism to share frequencies in the crowded two-meter band to minimize interference. Moreover, the ARRL pledges a donation of up to $500 to support the APRS QSY initiatives once the fund is established.

TAPR APRS SIG QSY Information Collection Questionnaire Survey

One of the first things started by the committee was a survey. The purpose of the survey was twofold. The first being a straw poll of the sentiment behind this issue. The second being the collection of information on who wanted to receive money from the QSY fund.

The survey was run from November 1st, 1997 until June 30th, 1998, at which time it was determined that saturation of the survey had resulted. Saturation being defined in this case as no significant change in the percentages (less than 2% over 3 months) shown in the survey results. The survey generated 486 entries of which 146 (30%) indicated digiowners, 253 (52%) indicated end-users, and 87 (18%) made no indication of status. The committee had hoped to reach over 150 wide digiowners with the survey and consider the 146 as a successfully reached goal.

All Respondents (486) - rank order

definitely	227	47%
willingly	94	19%
if everyone else does	90	19%
undecided	25	5%
definitely not	24	5%
maybe	18	4%
don't care	8	2%
not responding	0	0%

All Respondents combination (486) - rank order

definitely + willingly	321	66%
if everyone else does + don't care	98	20%
maybe + undecided	43	9%
definitely not	24	5%

Just looking at Wide Digi Owners (146) - rank order

definitely	69	47%
willingly	26	18%
if everyone else does	25	17%
undecided	13	9%
definitely not	7	5%
maybe	6	4%
don't care	0	0%
not responding	0	0%

Just looking at Wide Digi Owners combination (146) - rank order

definitely + willingly	95	65%
if everyone else does + don't care	25	17%
maybe + undecided	19	13%
definitely not	7	5%
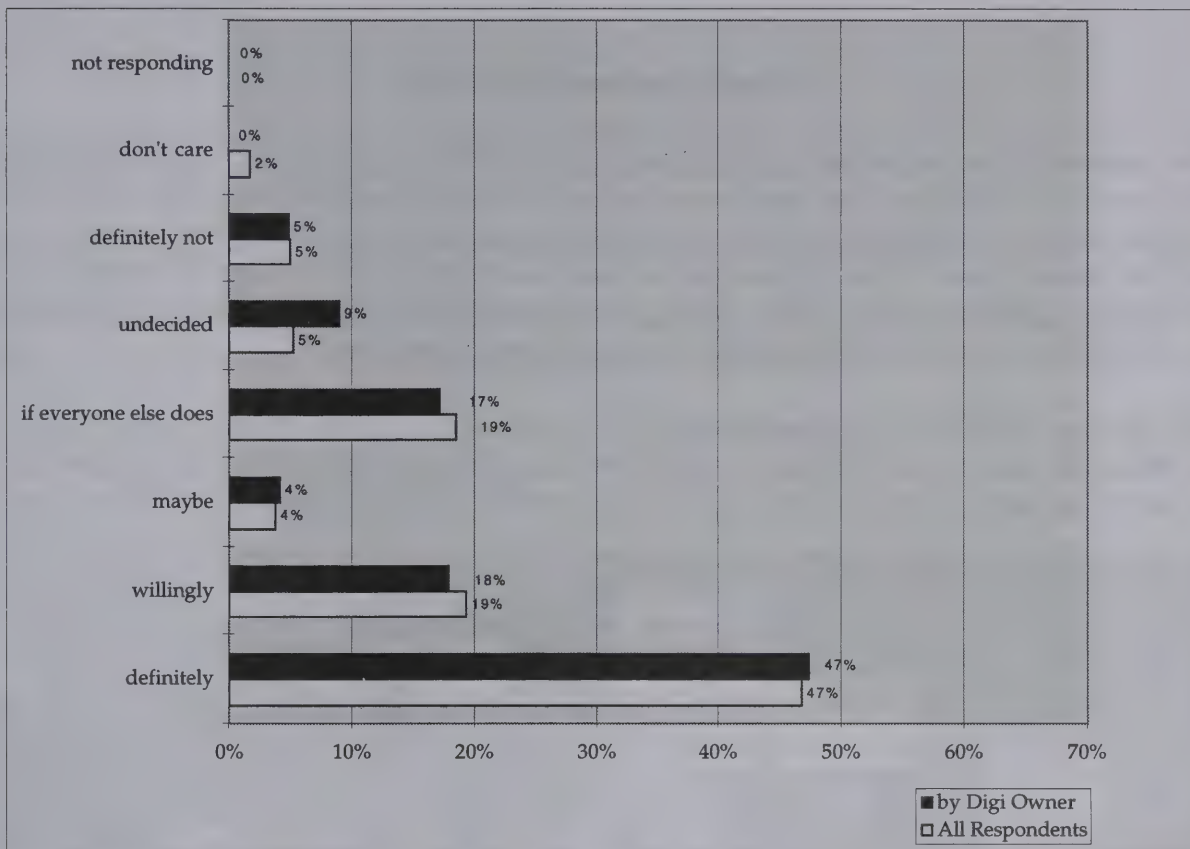
Graph showing the percentage of all respondents as compared to just Digipeater Owner responses to the survey. The information submitted by the total group and digipeater owner sub-group are nearly identical.
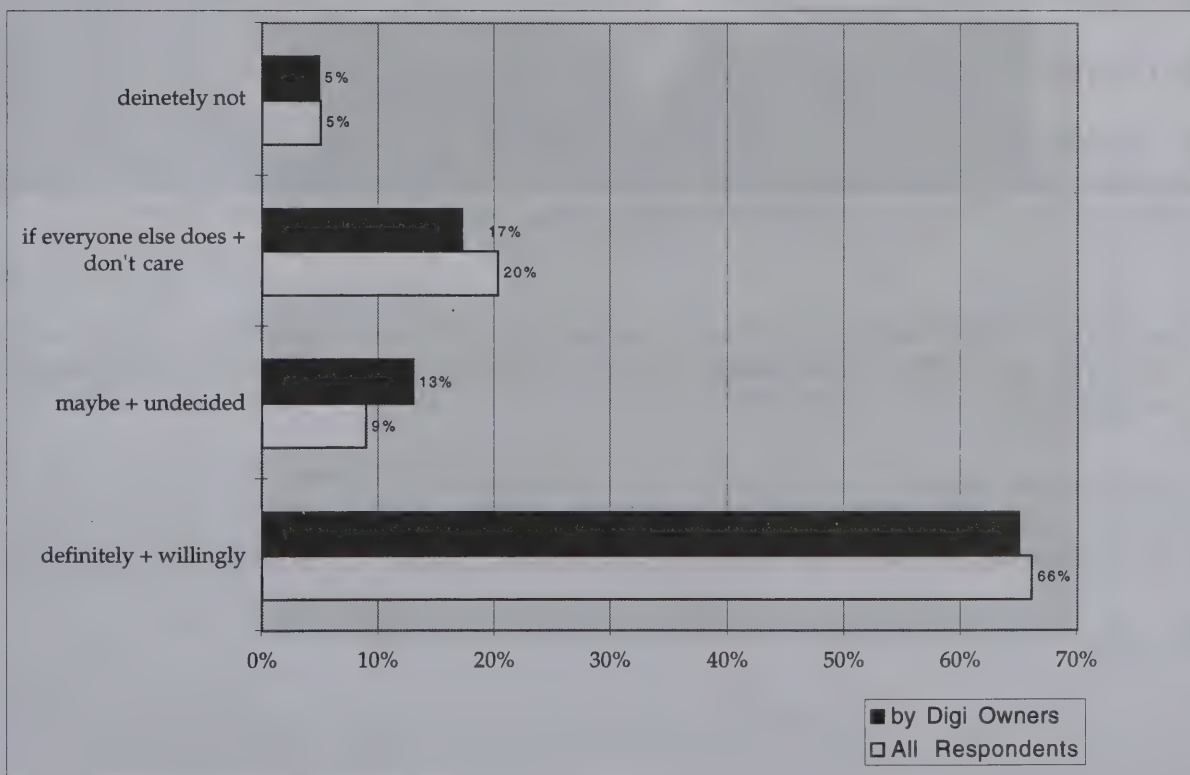

This graph shows the combination of like feelings about the potential QSY.

Current Status of APRS QSY

The position of the organizations involved (TAPR, AMSAT, ARRL) has always been that it is the choice of the individual ham whether or not to QSY, and this decision needs to be made on a local basis. It is not appropriate for one group of hams to tell another that they have to move, or when they should move. This applies just as much to one group of APRSers telling another, as it does to AMSAT telling APRS it has to move. Some areas have a tougher problem and need more time. Please be considerate of this, and try to help these situations.

Jeff Brenton, KA9VNV, has maintained a web page tracking the actual success of the APRS QSY. The following information and map is from his web page. Thanks to Jeff for allowing us to use some of his content in this article. (http://www.dididahdahdidit.com/APRSFreq.htm).
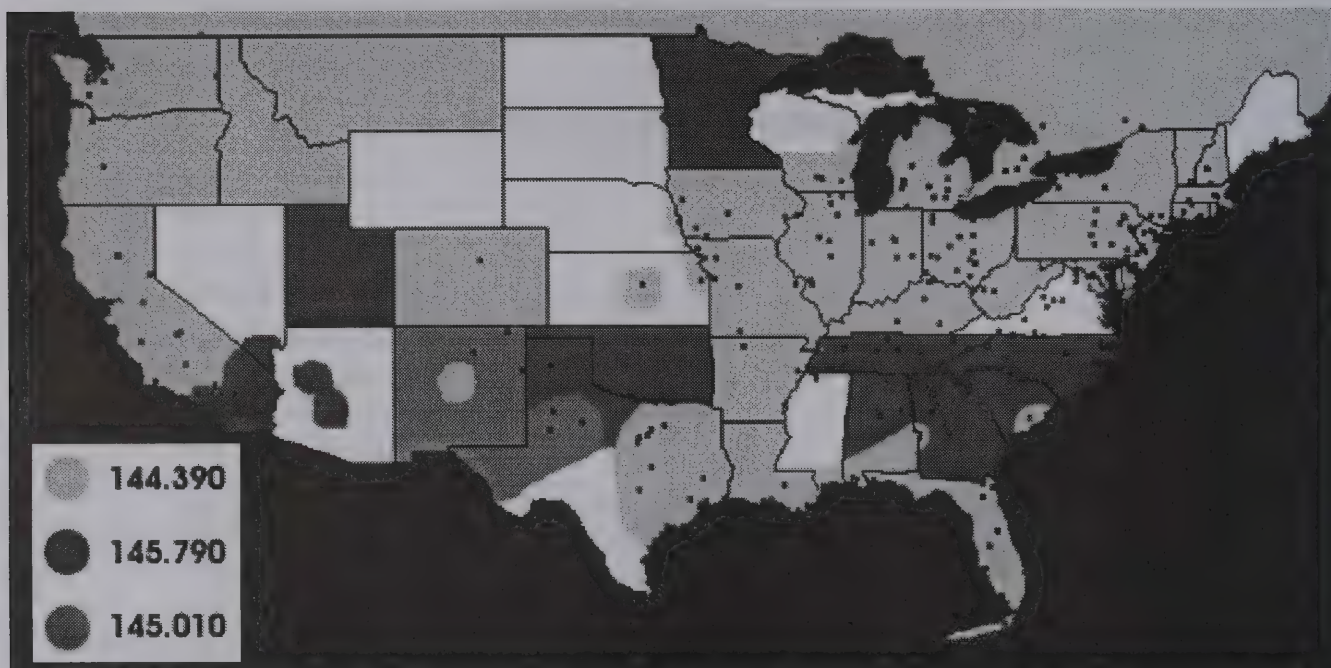
APRS QSY Frequencies as of June, 1998

The map lists the reported frequencies for various areas of the United States which will be in effect by early June, 1998. Some areas are in the process of switching to 144.390; others have switched already.

Jeff's page reports that:
- Two groups in Montana are establishing nets in the state, and it seems 144.39 will be the frequency of choice. The nets are being set up in/around Helena, and in the northwest, near Idaho.

- Oklahoma reports that the state will QSY by the end of the year.

- Northern California is committed to changing over all digis as they can be reached for maintenance. Due to uncertain weather conditions, this could take months for some of the more remote sites.

- Southern California, from the Los Angeles basin south, will remain on 145.79 until such time as coordination issues with many ATV stations can be resolved.

- Arkansas and the adjoining areas of Missouri are planning to change from 145.79 to 144.39.

- Virginia may be switching on an unspecified date. Reports are coming in that many Virginia stations have been seen on 144.39 during recent band openings.

- Ralph Fowler N4NEQ, reports that SERA (South Eastern Repeater Association) the official repeater coordination body for the states of Georgia, Kentucky, North and South Carolina, Virginia, West Virginia, Mississippi and Tennessee along with the Digis in Central and Northern Alabama, will be moving to 144.39 MHz on or about Halloween weekend (October 31) 1998. They have obtained recognition by SERA for APRS use of the new frequency, formerly allocated to AMSAT for future Oscar operations. SERA approves our use of this frequency with one major stipulation: We are to use proper engineering and RF practices to protect the 145.xx voice repeaters- whose inputs lie as close as 120 KHz from 144.39. Several digis are co-sited with these repeaters and others are very close. Our digipeaters must be outfitted with the proper filtration devices (band pass and notch cavities) to accomplish this task.

- Minnesota MAY be in for a change; rumor has it that Minneapolis/St. Paul will switch to 144.39 "around mid-June". Since this area has the majority of the Minnesota APRS activity, and users in the western portion of the state were asking about changing, can an official announcement be far behind? Such an announcement would affect the future of the north woods of Wisconsin, although there is not much reported activity there.

QSY Fund

In March of 1998 a message was sent to 19 individuals that had requested funding via the APRS QSY survey instrument. These 19 individuals represented only 3.9% of the 486 people submitting information to the survey. It seems that the vast majority of the APRS QSY has been self funded by groups and individuals.

The following 10 people requested $1265 worth of the $1500 collected from TAPR, AMSAT, and the ARRL. The remaining $235 is on hold to an 11th group, until the QSY change has been made.

N9QGS, Ron Malinowski, $30 for crystals.
N7ZEV, Frank Kostelac, $80 for crystals 2 digis in Las Vegas area
K7GPS, David Dobbins, $75 for recrystalling/tuning a digi in WA state
K5QQ, Jim Baremore, $70 xtals/tuning an NM digi.
WB0WNX, Dave Kaplan, $50 xtals/tuning Iowa digi
W3NRI, Greg Harbough, $50 xtals for two trackers
W9JBL, John Leonard, $90 DCI filter for Chicago wide
N4VDE, Ricky Davis, $25 for crystals for SC digi
KU0G, Jim Duncan, $300 for 10 radios.
KE4DGH, Tommy Ellison $495.00 for new radio and notch filter for co-site voice repeater

Conclusion

We should not underestimate the significance of the task accomplished by the amateur radio service and APRS community since the 1997 ARRL and TAPR Digital Communications Conference. In completing a QSY of this magnitude from the initial proposal to a shift of frequency on a national level which is widely accepted and implemented in such a short period of time is an event few have been successful in achieving in the history of the amateur radio service. While there are still areas of change to occur, progress in these geographical areas continues and we hope will eventually QSY over time.

The amateur radio service as a whole and the APRS community itself can congratulate itself for making the QSY happen and leaving the future frequency for ISS operations much less occupied and interference free for future astronaut communications to other amateur radio operators.

References
1. Frank Bauer, KA3HDO. 1997 ARRL and TAPR Digital Communications Conference. Web Page: http://www.tapr.org/aprsqsy/bauer/bauer.htm

2. Tom Clark, W3IWI. Tue, 14 Oct 1997. aprssig@tapr.org. Subject: Re: ***Important: APRS and AMSAT-Please Read

3. Full information on the Board meeting can be found at arrl.org. Refer to ARRL Bulletin 8 ARLB008 From ARRL Headquarters Newington CT January 20, 1998.

Back to the Packet Radio with MACA

Ikuko KAWAMURA[†]   Hideyuki ISHINAKA[‡]   Hiroshi MATSUNO[‡*]

[†]City Ascom Co.,Ltd.,   [‡]Faculty of Science, Yamaguchi University

* JH4CIN, matsuno@sci.yamaguchi-u.ac.jp,   1677-1, Yoshida, Yamaguchi, 753-8512 Japan

1 Introduction

Karn introduced MACA (Multiple Access with Collision Avoidancd) in [4] which was designed for packet radio network. It was used as the basis for the IEEE802.11 LAN standard. Thereafter, based on simulation studies of MACA, Bharghavan et al. fine tuned MACA to improve its performance and renamed their new protocol MACAW in [7].

In this paper, we first investigate the performance of MACA under the no hidden terminal situation. By an analytic way, we will compare the throughputs of MACA and CSMA[1]. We then review CSMA and some kind of protocols cosidered as extended versions of CSMA, and point out that MACA has an ability to get the throughput exceeding one. A suggestion in Conclusion in this paper will remind us that we are people who love amateur radio and have some interests in computers.

2 CSMA and Hidden Terminal

It is well-known that in Ethernet, CSMA/CD (Carrier Sense Multiple Access with Collision Detection) is used as a MAC protocol. When a packet to be transmitted by a station is occur, the station firstly sense the medium, then (1) if the medium is idle, it transmit the packet immediately or in accordance with some rule, (2) if the media is busy, it postpone the transmission. During the transmission, when the station detects a collision, it aborts its transmission, waits a random period of time, and then tries again. CSMA/CD is considered as an extended version of CSMA which was proposed by Kleirock and Tobagi [1] as a protocol on PRNs.

Figure 1 indicates the connection among terminals on packet radio network.

Teminals connected by a line in Figure 1 can comminucate with each other. Consider the case that the terminal B is transmitting a packet to the terminal D. In this situation, even if the terminal A tries to transmit a packet, it can stop to do that, bacause A can detect the packet from B. We should note that
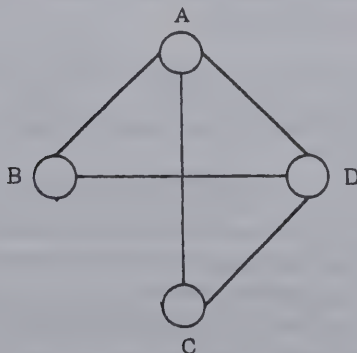
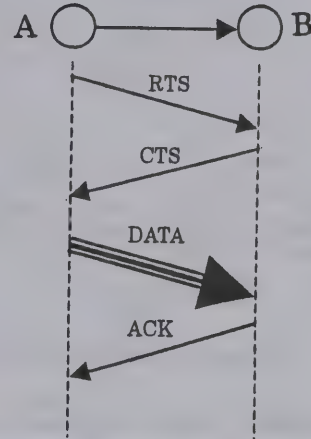

Figure 1: Connection among terminals

Figure 2: Sequence Diagram of MACA

there is no line between the terminals B and C. Thus, in spite of transmitting packet from B to D, C falsely conclude that it can transmit a packet. The packet cause collision at D with the packet from B to D. We call the terminals B and C "hidden terminal" each other. The existance of hidden terminal makes the throughput of PRN seriously decrease.

3 MACA

Karn proposed a new MAC protocol, Multiple Access with Collision Avoidance (MACA) as an alternative to the traditional CSMA protocol in [4]. One of the purposes of introducing MACA to a PRN is to eliminate the hidden terminal problem. Let us consider how the terminal A sends a packet to the terminal B in Figure 2. A starts the action by sending a short packet called RTS (Request To Send) packet to B. The RTS contains the length of the data frame that will eventually follow. Then B replies with a CTS (Clear To Send) packet which contains the data length (copied from the RTS frame). Upon receipt of the CTS frame, A begins to transmit a data. If B has received the data successfully, it sends the ACK [1] packet to A. The diagram of this sequences is shown in Figure 2.

Any station overhearing an RTS defers all transmissions by the time after the associated CTS packet would have finished. Any station overhearing a CTS packet defers for the length of the expected data transmission which was contained in the RTS and was copied to CTS.

Figure 3 shows the state diagram for MACA. A total of 8 states IDLE, CONTEND, WFCTS, WFACK, WFDATA,

---

[1]The ACK packet is introduced in [7] as one of the extended function of MACA.
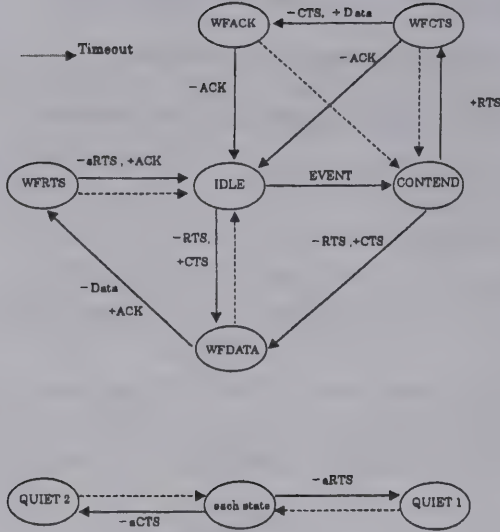
Figure 3: State Diagram of MACA

WFRTS, QUIET1, and QUIET2 exists, these states except WFRTS, QUIET1, and QUIET2 were presented in [7] in order to explain the transition rules in a concrete example.

We must need the state WFRTS since we have been considered the extended MACA by the ACK packet. In [7], the state QUIET was used in order to indicate the deferral rules on both of RTS and CTS. Recall that deferral times by RTS and CTS are different. In [7], the difference was realized by setting one of two values to a timer. Instead of using these two values, we adopt the two types of states QUIET1 and QUIET2 which corresponds to the deferral times by RTS and CTS.

When an event occurs (EVENT), the transition is done from the state IDLE to the state CONTEND. After the timer for contention is expired (dot-dash-line), a source terminal called A transmits an RTS (+RTS) and enters the state WFCTS. If a destination terminal called B accepts the RTS correctly, it responds to A by CTS after a time x. If A accepts the CTS correctly (-CTS), it transmits a data packet (+Data) after a time c, and then enters WFACK immediately. After a time d from when B recogizes the data packet was successfully accepted, it transmit ACK to A. Thereafter, if A receives ACK (-ACK) correctly, it enters IDLE immediately. It is a cycle in the case that the transmission is succeeded.

Note that the parameters x, c, and d which represent the times between RTS and CTS, CTS and Data, and Data and ACK, respectively, reflect the performance of a network node controller (NNC).

In one case when, in spite of sending RTS from the source terminal A, the destination terminal B can not send CTS, or the other case when B can not send ACK to A, A backs to the state CONTEND after the corresponding timer is expired (dot-dash-line).

Suppose that a terminal called C receives RTS from a terminal called D (-RTS) in a state IDLE. Then C transmits CTS after a time x (+CTS) and enters WFDATA state. After a time c from when D have received the CTS successfully, C begins to receive a data packet from C (-Data). If it is successfully

received, C sends ACK to D (+ACK) after a time d and enters WFRTS state. In the WFRTS state, if C receives the same RTS (-sRTS) as the one received before, then it transmits ACK (+ACK) and enters IDLE state. If C expires the timer for the RTS it simply enters IDLE state (dot-dash-line).

In each state, if a terminal overhears RTS (CTS) to be used for communications among another terminals (-aRTS (-aCTS)), it enters QUIET1 (QUIET2) and then keep quiet until the timer is expired.

We must note that because MACA does not perform carrier senses, we can neglect any hidden terminal situation in this protocol.

4 MACA in No Hidden Terminal Situation

It is interesting to compare the performances of MACA and CSMA. Because the CSMA is supposed to work on the situation with no hiddern terminals, considerations should be made on no hidden terminal situation.

We will now derive the throughput equation for the MACA. Since the technique for the derivation is similar to the one in [1], we only give sketches of that.

Let the "frame time" denote the amount of time needed to transmit the standard, fixed-length frame. Let us assume that the probability of transmission attempts per frame time is Poisson with mean G per frame time. The transmission attempts consists of newly generated frames and retransmitted frames that previously suffered collisions. We denote the ratio of maximum propagation delay to packet transmission time by $a > 0$. Further, in the following argument, we assume that the frame time is 1.

The expected value of the time needed to transmit a packet is simply the probability that no terminal transmit a packet during the time x between the arrival of an RTS and the departure of a CTS, it was noted in the previous section. Therefore,

$$\tilde{U} = e^{-xG}$$

Define a *busy period* to be the time during at least one station is not in an IDLE state and an *idle period* to be the time during all stations are in idle state. Let $\tilde{B}$ be the expected duration of the busy period and $\tilde{I}$ be the expected duration of the idle period. Then, the throughput is given by

$$S = \frac{\tilde{U}}{\tilde{B} + \tilde{I}}$$

Let P_{RTS0} denote the probability to be succeeded in transmitting a packet. It is easy to see that the probability P_{RTS0} is equal to the probability that during the time x no terminal transmits an RTS. Thus,

$$P_{RTS0} = e^{-xG}$$

And the period of time in which a cycle of transmission is completed is

$$B_{RTS0} = 4a + c + d + x + 1$$

, where c (d) is the time between the arrival of a CTS (a data packet) and the departure of a data packet (an ACK), it was noted in the perivious section.

On the other hand, the probability P_{RTS1} to be the packet is reserved is given by,
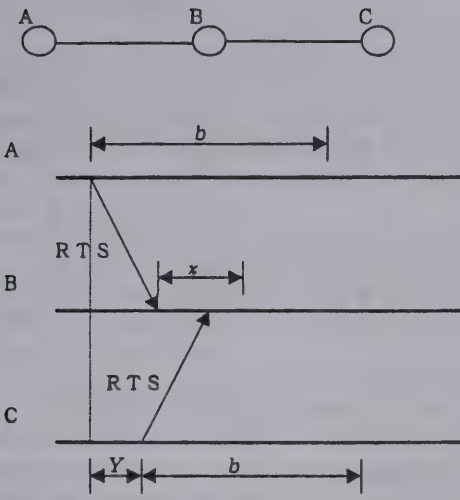
$$P_{RTS1} = 1 - P_{RTS0}.$$

Figure 4: Contention of RTS

We consider the case the number of stations transmit RTSs during the time x. Let $Y(< x)$ denote the time between the first station transmits an RTS and the last station transmit an RTS and let $\bar{Y}$ denote the expected value of Y. See Figure 4. Let b be the time needed to expire a timer for the state WFCTS.

In this case, the expected value of the busy period of time is given by

$$B_{RTS1} = \bar{Y} + b$$

The distribution function for Y is

$$
\begin{aligned}
F_Y(y) &\triangleq Pr\{Y < y\} \\
&= Pr\{\text{no arrival occurs in an interval } a - y\} \\
&= e^{-G(x-y)}, (y \geq x)
\end{aligned}
$$

The average of Y is therefore given by

$$Y = x - \frac{1}{G}(1 - e^{-xG})$$

Thus, we have

$$B_{RTS1} = b + x - \frac{1}{G}(1 - e^{-xG})$$

Then, the expected duration of the busy period is obtained as follows.

$$
\begin{aligned}
B &= P_{RTS0}B_{RTS0} + P_{RTS1}B_{RTS1} \\
&= (4a + c + d - b + 1)e^{-xG} + b + x \\
&\quad - \frac{1}{G}(1 - e^{xG})^2.
\end{aligned}
$$

From the argument above and the average duration of an idle period is simply representable as $\bar{I} = \frac{1}{G}$, we can get the throughput equation as follows.

$$S = \frac{Ge^{-xG}}{G\{(4a + c + d - b + 1)e^{-xG} + b + x\}} \\ - (1 - e^{-xG})^2 + 1$$

We must note that all of $a, c,$ and d in the above fomura are constants, because of the no hidden terminal condition. Then, we will later observe the behavior of the throughput with respect to only the variable x.
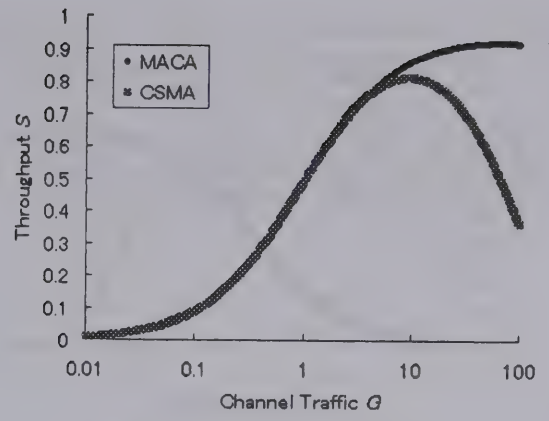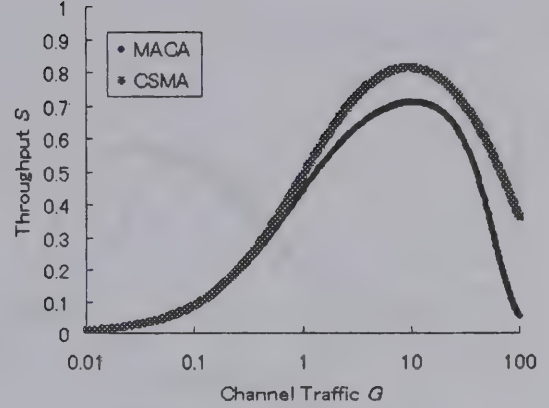

Figure 5: $a = 0.01, x = 0.005$

Figure 6: $a = 0.01, x = 0.01$

On the no hidden terminal condition, if the performance of MACA is better than that of CSMA, we can conclude that MACA has an inherently good performance than CSMA. The throughput equation of CSMA was given in [1] as follows:

$$S = \frac{Ge^{-aG}}{G(1 + 2a) + e^{-aG}}.$$

We will investigate the difference between the throughputs of MACA and CSMA by changing conditions of the delay time a and the time between RTS and CTS. Figure 5 to 8 show the results.

It is well-known that the throughput of CSMA decreases according to the increase of the channel traffic. On the other hand, it is clealy evident from Figure 5 that even in high channel traffic, the throughput of MACA does not decrease. The reason to be the phenomenon occured is that although when a collison occurs, the time between one frame time and two frame time is lost in the CSMA environment, in the MACA environment the time to be lost is only the short frame times to be used by RTS and CTS.

We will compare Figure 5 and Figure 6. It is no wonder but, while the throughput of CSMA does not depend on the parameter x, the throughput of MACA does. It follows that in no hidden terminal and low propargation delay environment, in order to overcome the CSMA, the performance of RTS and CTS interchange at MACA should have excellent efficiency.

By comparing Figure 5 and Figure 7, we can find that the MACA is less sensitive to increases in the delay a, as compared to the CSMA.

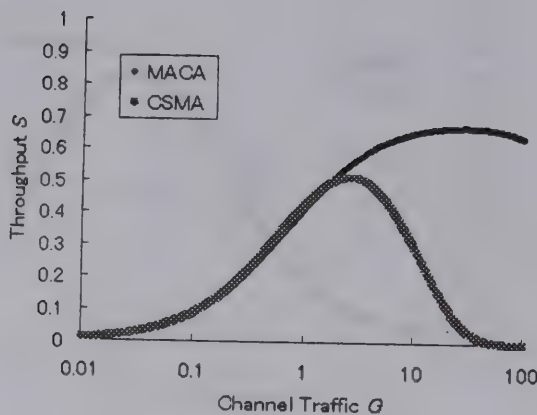Figure 8 indicates that in the low channel traffic the through-
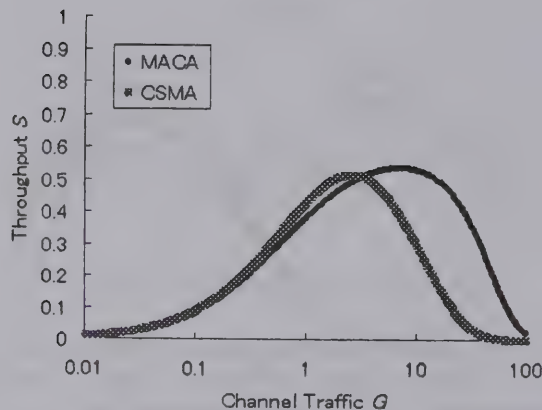
Figure 7: $a = 0.1, x = 0.005$

Figure 8: $a = 0.1, x = 0.01$

put of CSMA is better than that of MACA, but in high traffic the positions of these protocol are reversed.

5   MACA and Other Protocols

Karn also pointed out in [4] that less well-known than hidden terminal but a serious problem for CSMA protocol is the problem of exposed terminal. Let us consider the situation that the terminal B is transmitting to the terminal A as shown in Figure 1. If the terminal C senses the medium, it will hear an ongoing transmission and falsely conclude that it may not send to the terminal D. But, in fact, the packet transmitting from C to D gives no conflict at A. We can find that the existance of exposed terminal should lose the chance to transmit more than two packets simultaneously.

It is noted that if the whole period of time is completely occupied by packets, then the throughput is defined to be one which is the upper bound of the range.

Many investigations about the MAC protocols each of which can be considered as an extended version of CSMA are performed [2, 3, 6, 11, 9]. The concepts of all of these protocols are "How do we overcome hidden terminals on the protocol with carrier sense ?". We call these protocols "carrier sense type protocols". The key of carrier sense type protocols is to locate a central station on a PRN which informs the presences of transmitting terminal to all terminals by a tone signal. Thus, if a terminal having a packet to be transmitted receives the tone signal, it is going to postpone the transmission. This mechanism leads the terminals to preserve collisions, but for even faraway terminals from the transmitting terminal (above two
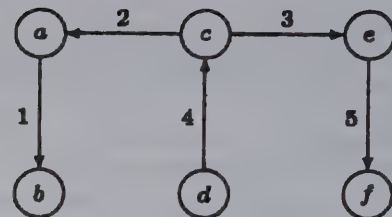

Figure 9: A scene on a PRN

hops), the transmission is going to be reserved. In other words, if we use the carrier sense type protocol, all terminals in a PRN necessarily should be exposed terminals. This is stupid, because, obviously, these above-two-hops-away-terminals can transmit data independently.

On the other hand, MACA uses RTS and CTS as a mechanism to aviod collisions of packets. Moreover, MACA is worked on a basis of connecting information up to two hops around some fixed node. It is intersting to compare the fact above and the fact that in order to work carrier sense type protocol in a PRN well, any terminal must care all terminals in the PRN. Then, by introducing MACA to PRN, we can get efficient throughput exceeding one. As concerning to carrier sense type protocols, we never get the performance as the throughput is exceeding one, because the protocol has no ability to achive the simultaneous transmission.

We should note that there are some cases that some terminals within two hops can transmit data simultaneously. The PRN in Figure 9 has six terminals a,b,c,d,e, and f. An arrow between two terminals indicates the flow of data. Each of these six arrows is labeled by an integer. We can easily find that the following conbinations of arrow are available for simultaneous transmission.

$$(1,3), (1,4), (1,5), (2,5), (4,5), (1,4,5)$$

Note that the origins of arrows of each of conbinations above are within two hops. An algorithm to get simultaneous transmission is presented in [10].

Another Protocols to be enabled simultaneous transmissions had already introduced in [5, 8]. It is similar to MACA that both of the protocol STS[5] and STMA/DA[8] use two types of short frames or tones such as RTS and CTS of MACA to be simultaneous transmission available. In addition to this, STMA/DA uses special tones for avoiding collisions. Moreover, at both of STS and STMA/DA, directional antennas are used to increase the throughput by spatially reusing the channel.

In order to show efficiency of the proposed protocol, the comparison of throughput between that and carrier sense type protocol was performed in each of [5] and [8]. According to expectation, the proposed protocol has more efficient performance than the carrier sense type protocol. But it is a natural result, because the comparisons between the protocols which can simultaneous transmissions and can not are made. It is our present interest to compare the performances among these protocols to be able to simultaneous transmission such as MACA, STS, STMA/DA, and others.

6 Conclusion

CSMA, which is the origin of CSMA/CD, had been proposed by Kleinrock and Tobagi as a MAC protocol on Packet Radio Network (PRN) in 1975 [1]. They had noticed that the serious problem of CSMA is the existence of hidden terminal, and had shown a way of solution to the problem in the paper [2] just followd by [1].

Nowadays, CSMA/CD is a famous protocol on a bus network such as Ethernet. Needless to say, since there is no hidden terminals in a bus network, CSMA/CD works well. But, actually, in a PRN, it is reasonable to consider that no hidden terminal situation is a special case.

Karn had proposed in [4] that "Let's ignore data carrier detect (i.e., carrier sense)". This suggestion is not only eliminate a bad effect of hidden terminal, but also leads to the efficient communication way in a PRN, that is, "simultanious transmission'.

Recall that, CSMA had been originally arised as a protocol for Packet Radio. Now, our turn has come again. Let's "Back to Packet Radio with MACA" for more excellent communications.

References

[1] L. Kleinrock and F.A. Tobagi; Packet switching in radio channels: Part I − Carrier sense multiple access modes and their throughput delay characteristics, *IEEE Trans. Commun.*, vol.COM-23, no.12, pp.1400-1416, 1975.

[2] Tobagi, F.A. and Kleinrock, L.: Packet switching in radio channels: Part II − The hidden terminal problem in carrier sense multiple-access and the busy-tone solution, *IEEE Trans. Commun.*, vol.COM-23, no.12, pp.1417-1433, 1975.

[3] Mukumoto, K. and Fukuda, A.: Idle signal multiple-access (ISMA) scheme for terrestrial packet radio networks, *Trans. IECE of Japan*, vol.J64-B, no.10, pp.1107-1114, 1981 *(in Japanese)*.

[4] Karn, P.: MACA − A new channel access protocol for packet radio, *ARRL/CRRL Amateur Radio Ninth Computer Networking Conf.*, pp.134-140, 1990.

[5] Yun, T.S. and Hung, K.W.:Design algorithm for multihop packet radio networks with multiple directional antennas stations, *IEEE Trans. Commun.*, vol.40, no.11, pp.1716-1724, 1992.

[6] Shigeno, H., Ohnishi, Y., Yokoyama, T., and Matsushita, Y.: Performance evaluation of CTMA as a solution of the hidden terminal problem, *Tech. Rep. IEICE of Japan*, SAT92-18 (IN92-17), 1992 *(in Japanese)*.

[7] Bharghavan, V., Demers, A., Shenker, S., and Zhang, L.: MACAW: A media access protocol for wireless LANs, *Proc. SIGCOMM'94 Conf.*, ACM, pp.212-225, 1994.

[8] Shigeno, H. and Matsushita, Y.:Desgin and performance evaluation of new peer-to-peer MAC scheme STMA/DA for wireless LAN using directional antennas, *Trans. IEICE*, vol.J-79-B-I, pp.41-50, 1996 *(in Japanese)*.

[9] Zdunek, K.J., Ucci, D.R., and LoCicero, J.L.: Packet radio performance of inhibit sense multiple access with capture, *IEEE Trans. Commun.*, vol.45, no.2, pp.164-167, 1997.

[10] Minami, T., Someya, K., and Matsuno, H., Time division scheduring problems on packet radio networks, Tech. Rep. IEICE, COMP97-24, 1997.

[11] Matsuno, H., Ebisui, T., and Ando, H.: Effect of an extra ability to central station in CTMA, *Trans. IPS Japan*, vol.39, no.4, pp.1049-1057, 1998 *(in Japanese)*.

Spread Spectrum in the Amateur Radio Service: Current Status and Historical Notes

Barry McLarnon, VE3JF
Dewayne Hendricks, WA8DZP

Amateur SS in the USA

The beginnings of amateur SS experimentation date back to late 1980, when Paul Rinaldo and a few others in AMRAD formed an SS Special Interest Group. The group decided to seek an STA from the FCC which would allow SS experiments to take place in some of the amateur bands, and they found a receptive audience at the Commission. Thanks in large part to the urging of Mike Marcus of their Office of Science and Technology, the FCC was interested in initiatives that would help SS technology make the move from its military roots into commercial applications. The STA was granted in March 1981, permitting FH experiments in the 80, 40, 20 and 10m HF bands, DS in the 420-450 MHz band, and SS EME experiments. About 30 amateurs were named in the initial STA. An amendment permitting FH experiments in the 144 MHz band was added later in the year. The group initially focused on the HF bands, and then on the 144 MHz tests. The emphasis on FH is not surprising, given its roots in narrowband technology. Much of the work involved experimentation with various synthesized amateur rigs to see which could be effectively frequency-hopped, and construction of controllers to perform the hopping and synchronization. These were early days as far as amateur digital communications were concerned, so these initial efforts concentrated on application of SS to analog voice transmission.

Again showing its interest in fostering non-military SS development, in late 1981 the FCC proposed to change the rules of the amateur service to permit SS operation (for Advanced and Extra class licensees only) in the VHF (50, 144 and 220 MHz) bands. The proposal was not met with great enthusiasm in certain quarters, however, and it slipped onto the back burner. This was the beginning of a fairly lengthy hiatus on the regulatory front, but some experimentation continued. Things began to heat up again in 1984, when the FCC granted a second STA to AMRAD, and also let it be known that it was again considering rule changes to open SS experimentation to all US amateurs, in the VHF bands only. In May 1985, the new rules were unveiled, and, lo and behold, an about-face had taken place: SS was to be permitted, but only above 420 MHz! Much of the opposition to the initial idea of permitting SS in the VHF bands no doubt came from within the amateur community (especially where the 144 MHz band is concerned!). In addition, there was concern expressed in some quarters (such as the National Association of Broadcasters) about the potential for interference to TV receivers. As far as 220 was concerned, there were already rumblings that commercial interests had designs on the band, so it's not surprising that it was dropped from further consideration for amateur SS work. Actual implementation of the rule changes was delayed for one year; in the meantime, the ARRL formed an ad hoc committee to consider standards for amateur SS. Later in 1985, the FCC reinforced its new-found commitment to not allow amateur SS operation at HF and VHF by abruptly canceling AMRAD's STA.

The new FCC rules went into effect on June 1, 1986, beginning a new era of access to SS

experimentation for all US amateurs (albeit one with a number of restrictions). The highlights:

- both DS and FH permitted, but only three different spreading codes authorized
- above 420 MHz only
- 100W PEP maximum transmitter power
- all transmissions must be logged
- domestic communications only
- transmissions to be ID'ed by means decodable with narrowband receivers

Unfortunately, this noteworthy event went by virtually unnoticed by amateurs. A small core group in AMRAD, led by N4ICK and N4EZV, continued SS experiments, but were not joined by a host of new recruits. Although details about much of their SS hardware were published, duplicating the designs presented a daunting task for most amateurs.

Although the new rules were a major step forward for SS in amateur radio, some of the restrictions presented obstacles to serious experimentation, particularly the lack of access to the VHF bands and the choice of only three spreading codes. The latter restriction eliminates the possibility of experimenting with CDMA techniques. It also means that most of the chipsets and other commercial spread spectrum hardware developed in recent years could not be used for amateur applications. This led Bob Buaas, K6KGS, who had participated in the previous AMRAD STA, to request a new STA. The new STA, granted in 1992, removed the spreading code limitations, permitted experiments in the VHF bands, and also allowed use of hybrid DS/FH modulation techniques. The requirements for logging and ID'ing of transmissions remained. The STA was renewed the following year, and in 1994, it was renewed for an indefinite period.

Now, let's fast-forward to the end of 1995. Nearly ten years after the rules were changed to permit amateur SS experiments above 420 MHz, and fifteen years after the first SS STA was issued, there were still only a small handful of amateur experimenters working with SS. In an effort to change this, the ARRL petitioned the FCC for some additional rule changes, the purpose of which were to "facilitate, to a greater extent than is done by the present rules, the contributions of the Amateur Service to the development of spread-spectrum communications". The major changes requested were to:

- drop the restrictions on spreading codes and permit hybrid DS/FH emissions
- permit SS communications with amateurs in other countries where SS emissions are permitted
- add a requirement for automatic power control when transmitter powers of more than one watt are used

Notably absent from the petition are any requests to relax the logging and ID requirements, or to extend SS experimentation to the HF and VHF bands. The petition also stipulates that SS transmissions be "brief".

In its comments on the ARRL petition, TAPR was generally supportive, but urged the Commission to go further in relaxing the SS rules. TAPR proposed that the ID requirements be dropped completely, that SS tests not be restricted to "brief" transmissions, and that SS be permitted in the VHF bands covered by the Buaas STA (plus the new 219-220 MHz allocation). TAPR also commented that the automatic power control provisions should be phased in over a period of time rather than taking effect immediately. Other commenters also took up the theme that the the Buaas STA become the basis for the

rule changes. Some felt that if power control is to be mandated, it should apply to all services and not just SS. And, not surprisingly, there were a number of negative comments filed by the weak signal and repeater community. The full text of most of the comments and reply comments can be found on TAPR's web site.

Given the controversial nature of the ARRL petition, it appeared that a resulting NPRM from the Commission might be some time in coming. Consequently, TAPR filed a petition in April 1996 to have the Buaas STA extended to its membership. The ARRL, steadfast in its opposition to SS in the VHF bands (with the exception of 219-220 MHz), filed comments objecting to that aspect of the STA request, despite the fact that those bands are already accessible to any US amateur SS experimenters who join the Buaas STA. After a series of discussions took place between TAPR and the ARRL, the League removed their objections to the TAPR petition and the FCC granted TAPR an STA on November 6, 1996. Information on the TAPR STA and how to join it can be found at http://www.tapr.org/ss/.

On March 3, 1997 the FCC issued WT Docket 97-12 which basically incorporated all of the ARRL's proposal. Comments, which were due sixty days later were filed by several amateur radio organizations, a few individual hams, and some non-amateur radio groups. Most of the comments and reply comments filed in this proceeding are available on the TAPR website at the URL cited earlier. Basically, the comments fall into three general categories: 1) Those who approve of the proposed rules as is or who wants the rules loosened up even more, 2) Those who approve in concept, but who want to see the rules reflect more protections for other modes, and 3) Commercial interests who don't want interference from hams using SS who are sharing spectrum, even though we're a licensed service and they are

not. The comment/reply comment period for the NPRM closed on June 5, 1997. As of the writing of this article, the FCC has yet to issue a Report and Order stating just what the new SS rules are be. In the meantime, TAPR continues conduct SS experiments under its STA and submits reports to the FCC every six months documenting its results. Last year, TAPR embarked on a major effort to develop a 900 MHz SS radio that could be made available to its members. Details on the project can be found at http://www.tapr.org/tapr/html/taprfhss.html. As of the writing of this article, the project is still on-going.

Amateur SS in Other Countries

Thanks to rule changes which took place a few years ago, Canada's amateur radio service is largely deregulated. Detailed regulations defining subbands and permitted types of emissions have been replaced simply by bandwidth limits. On the downside, these limits are too low in the bands below 430 MHz to permit effective SS transmissions, except at very low data rates. The maximum bandwidths are 6 KHz in the HF bands below 28 MHz (except the 10.100-10.150 MHz band, where it is 1 KHz), 20 KHz at 28-29.7 MHz, 30 KHz at 50-54 and 144-148 MHz, 100 KHz at 220-225 MHz, 12 MHz at 430-450 and 902-928 MHz, and no limit other than the band edges in the higher bands (1240-1300 MHz, 2300-2450 MHz, etc.). The ID requirements simply state that callsigns be transmitted at the beginning and end of each "period of exchange of communication", and at intervals of not more than thirty minutes during these periods. Maximum transmitter carrier power is 750 W for holders of Advanced Class certificates, and 190 W for Basic Class. So, at UHF and above at least, Canadian amateurs have considerable latitude for SS experimentation. The regulatory agency (Industry Canada) has no mechanism comparable to the

STA in the US for granting waivers to the existing rules, so it may be difficult for Canadians to participate in SS experimentation in the lower bands.

We haven't heard much about amateur SS experiments taking in place in other parts of the world. It is probably safe to say that the regulatory atmosphere in most countries regarding the amateur service is less permissive than in the US and Canada. One exception is Israel, where experimentation is strongly encouraged, and there are few restrictions on emission modes and data communication protocols. Some experimental work with FH equipment is currently taking place. A similar attitude towards experimentation prevails in the UK, although the amateur regulations do not yet permit SS transmissions. The main sticking point has been in determining a method by which the spreading sequence being used can be made known to stations monitoring the SS signals. Discussions are continuing, and it seems likely that amateur SS in some form will be legal in the UK before too long.

What We've Learned So Far

Like most programmers, who love to write code but hate to document it, SS experimenters have not done a great job of publishing the results of their work. What follows is therefore not a comprehensive summary of the results to date, but simply some comments based on a few publications, STA reports and private communications.

Much of the early amateur SS work has focused on the adaptation of conventional narrowband amateur gear to SS. What this boils down to is controlling the synthesizer of an analog FM or SSB voice radio to cause it to frequency-hop, plus providing a means of synchronization. This has been accomplished with some degree of success, but the synthesizer

implementations in these radios are clearly sub-optimal when it comes to frequency hopping. This leads to relatively slow FH systems, which in turn increases both susceptibility to interference of the FH system and the severity of interference to other services. In particular, the channel dwell times were long enough to key up repeaters in some tests. This problem was dealt with simply by reprogramming the synthesizer controller to avoid hopping onto repeater input frequencies.

These early SS implementations were fairly rudimentary, and it is unlikely that slow FH analog voice transmissions have a great future in amateur radio. Nevertheless, they provided a good demonstration that a working form of SS could be accomplished with simple equipment, coupled with some amateur ingenuity. This work helped to demystify SS; more importantly, it showed that even a low-end SS system could be operated with minimal interference to existing services.

In more recent work, attention has shifted to SS data communications. Under the Buaas STA, experimental equipment for DS, FH and hybrid DS/FH transmission and reception was constructed and tested in several VHF and UHF bands. Data rates ranged from 12 Kbps to 0.5 Mbps, over ranges of up to 50 miles. These tests again demonstrated that SS could coexist with narrowband emissions in the amateur bands without causing significant interference to those activities. On the other hand, the performance of the SS systems was considerably hampered by the high-power narrowband transmitters. One conclusion that can be drawn from this is that in order to take full advantage of SS, particularly in the bands below 450 MHz, the SS systems will need to be quite sophisticated (compared to, for example, the current crop of Part 15 devices). Techniques such as the use of very high processing gain, adaptive frequency hopping, forward

error correction (FEC), notch filtering, automatic power control and new philosophies for determining subband allocations will become important keystones in making SS work. Another conclusion is that power control (using no more power than is necessary to maintain communications) should be practiced by ALL users of the amateur bands.

In addition to the work mentioned above, there are a number of people in the amateur community who have extensive practical experience with ISM band SS devices and other commercial SS hardware. They have shown that excellent performance can be obtained over considerable distances with properly-engineered RF links. Their experience and expertise will be invaluable in helping SS to take its rightful place in amateur radio communications.

The Way Ahead

More than 15 years have passed since the beginning of amateur SS work, and yet it has attracted only a small handful of intrepid experimenters. This is indicative of the changes which have taken place in the amateur radio hobby over the past few decades: few hams build their own radio hardware anymore. Even for those inclined towards hardware construction, building a working SS system from scratch is a daunting task. SS will clearly not become a significant activity in amateur radio until kits or ready-to-run RF modem hardware becomes available. Like the TNC before it, this is a breakthrough which TAPR would like to facilitate. RF projects are always difficult to complete successfully, but the enormous commercial interest which has developed in PCS and wireless LAN systems lately will be a great help. Many components, modules and chipsets developed for commercial applications can be applied to amateur SS development. Moreover, the numerous ISM band SS modems now available in the marketplace provide an easy means

of getting one's feet wet in SS. Although a few amateurs have been using these devices for some time, many others have been unaware of their existence. The cost has also been an impediment; until recently, nothing was available for under $500, which might be considered the "pain threshold" for radio purchases. Now, however, it is possible to get some of the devices for well under this figure, and some of the more mature products are occasionally becoming available at deep discounts. For example, a batch of 900 MHz WaveLAN cards recently came on the market for only $200 each. Think of it – this is a device that does 2 Mbps, CSMA/CA media access, and includes the antenna (dual-diversity, short range), radio, modem and ISA bus computer interface. There are packet, NDIS, ODI and Linux drivers available for the card. Now consider what most hams using 9600 bps packet (less than 1/200 the raw data rate of WaveLAN) have invested in their equipment – the mind fairly boggles at the comparison!

Up until now, the amateurs using these ISM band devices have simply operated them as unlicensed devices under Part 15 (or the equivalent outside the US). However, if the regulations permitted operating them in the amateur service, then we could overcome the ERP limits imposed on unlicensed operation and use high-gain antennas to increase their range. Since modifying the frequency of operation of most of these devices is probably quite difficult, it is fortunate that there is considerable overlap between the UHF ISM bands and the amateur bands. In North America, the 902-928 MHz ISM band coincides exactly with the 33 cm amateur band, so operation of ISM SS devices in the amateur service is straightforward, provided that ID and other regulatory requirements can be met. The 2.4 GHz ISM band, on the other hand, runs from 2400 to 2483.5 MHz (in North America – the band for unlicensed operation varies in other parts of

the world), whereas the amateur segment stops at 2450 MHz. For some devices, this presents no problem; for example, 2.4 GHz WaveLANs are DS units with about 22 MHz bandwidth and several choices of center frequency which can be programmed with a software utility supplied by the manufacturer. There are several choices of center frequency which confines the emission to the 2400-2450 MHz portion of the ISM band, making it a candidate for amateur experimentation. FH units operating at 2.4 GHz present more of a problem, since they generally use nearly the full ISM band. They are required to use at least 75 non-overlapping channels, so for the higher bitrate units at least, they must use most of the band. However, the hopping sequences and center frequencies are usually programmable, so it is generally possible to create an amateur band version of the FH units without any hardware modifications. This must be done by the manufacturer, however; they are understandably reluctant to release the reprogramming software to users. One of the authors (McLarnon) has experimented with RangeLAN2 hardware which has been programmed by Proxim to hop within the 2400-2450 MHz amateur allocation. The other author (Hendricks) has also experimented with a wide range of Part 15 devices. The details of that work can be found at http://wireless.oldcolo.com.

Make no mistake, the ISM band units are not the ultimate in SS technology. Most of them have quite low processing gain (especially the higher-speed systems), and none of them have automatic power control. They are not designed for building CDMA networks, and their tolerance for narrowband interference tends to be quite limited (especially in the DS systems). You probably won't find exotic features such as RAKE receivers and adaptive hopping patterns. Nonetheless, these are very useful devices which could play a very significant role in amateur packet networking.

The times are a-changing in amateur packet radio. Once upon a time, there was a dream of nation-wide and even worldwide networks, all connected by radio links. This didn't seem like too farfetched a dream when the major packet application was the BBS and store-and-forward movement of mail and bulletins. The amazing rise in popularity of the Internet has changed all that. Instead of the stodgy old BBS interface, Internet users have fallen for the seductive charms of multimedia web browsing, real-time conferencing, digital audio and video, Usenet news, mailing lists, etc. Traditional packet radio pales by comparison (especially at the "standard" bit rate of 1200 bps!). Although some still pursue the idea of wide area low-speed BBS-based radio networks, most of the "best and the brightest" have drifted away to more rewarding pursuits. The truth is, building a wide area packet radio network with enough bandwidth to support the applications that most people now want to use is simply unrealizable - radio amateurs do not have the resources nor the collective will and coordination to pull it off. Does this mean amateur packet radio is doomed to wither and die? Not necessarily. The new paradigm which is starting to emerge in packet radio is the coupling of the TCP/IP-based applications of the Internet with the building of higher-speed packet radio MANs. In other words, think globally, but act locally. Rather than expend your energy on long-haul radio links which are difficult to build and maintain, concentrate on putting up as much local bandwidth as you can muster, and then link your MAN to other areas of similar activity by the most expedient means available. This may be a radio link, or landline, or satellite - but the easiest means is usually the establishment of an Internet gateway. When radio-based MANs offer speeds substantially greater than landline modems can deliver, packet radio starts to look interesting again. Add to this the possibility of mobile and

portable operation, and you have a recipe for putting the excitement back in packet radio.

So, here's the scenario: we need the capability to transmit data at speeds ranging from, let's say 56 Kbps at a minimum, to T1 rates or more. The radio links will be short or medium range (up to 50 km?) and will be predominantly in urban environments. For maintainability, most of the nodes of the network will be at users' homes, apartment and office buildings, etc. Some of them will be mobile. Multipath will nearly always be present to some degree, and interference from other services and other amateur operations are very likely. Our network must continue to perform well in the face of such adversity, and it should cause minimal disruption to the other services. Of course, it should carry all of the Internet-type applications transparently, and work with all of the popular computer hardware and operating systems. This may sound like a pipe dream, but it is actually very doable. The magic ingredient needed to make it happen is - you guessed it - SS modem technology. The current generation of ISM band SS devices can take us a long way towards this goal. To take us the rest of the way, development work spearheaded by TAPR will be aimed at higher-performance SS systems, with capability to use other amateur bands in addition to 900 MHz and 2.4 GHz.

Resources

To get plugged into what's happening with SS in amateur radio SS, your starting point should be the TAPR Amateur Radio Spread Spectrum Communications page at

http://www.tapr.org/ss/index.html. You'll find updates and historical notes on the STAs and other initiatives on the regulatory front, the current SS rules in the US, tutorial information, and links to other SS resources on the net. Elsewhere on the TAPR web site, you can find out how to sign up for the SS SIG/mailing list (hint: send email to listserv@tapr.org with

subscribe ss YourName

as the first line of text). This mailing list is intended to be the place for SS experimenters to exchange information, talk about new products, plan tests, etc. If you're interested in SS applications in the HF bands, you should also join the TAPR HFSIG. For information on the many ISM band SS modem products, check out VE3JF's survey at <http://hydra.carleton.ca/info/wlan.html>. It includes summary tables of the product specifications and links to the manufacturers' web pages, product reviews, and other relevant references. As far as Usenet news is concerned, there is no newsgroup dedicated to SS technology, but discussions about SS wireless LAN hardware sometimes show up on the comp.std.wireless newsgroup.

Printed reference material is somewhat more limited. The definitive guide to the first ten years of amateur radio SS work is the ARRL Spread Spectrum Sourcebook. It also includes useful tutorial information, and some construction details for determined homebrewers. Beyond that, you'll have to hunt down journal articles and textbooks - again, see the references in TAPR's web pages. The books by R.C. Dixon are quite readable, with a minimum of math.

Current Status of Amateur Spread Spectrum Radio in Japan

Katsuhiko Morosawa, 7K1NCP/JH0MRP/KD5EYI

Packet Radio User's Group (PRUG)

P.O. Box 66, Tamagawa, Setagaya-ku, Tokyo 158-0094 JAPAN

Jh0mrp@prug.org

Abstract

We have practiced field tests using our PRUG96 system with SS transceivers, where we performed the 30km distance QSO. The output power of the transceiver is only about 30mW and the front gain of the antenna is 21dbi. We also confirmed that our system enables us to use the internet applications ('web browsing', 'videoconference' and so on) with practical speed.

Key words: PRUG96, Spread Spectrum, SS, PS, IPSM

Introduction

In 1997, some Japanese amateur radio stations were licensed Spread Spectrum (SS) by the MPT, which wasn't allowed before. We, the PRUG96 members applied for SS licenses unifying our method to the same one and practiced field tests three times, while most of other stations were not able to communicate each other, since each of them used different SS methods. In this paper, the author would like to explain these results of our experiments, and mention the state of alpha/beta tests just in progress right now.

The first experiment between Kitakyushu and Shimonoseki (Nov. 2, 1997)

On November 2, 1997 --- just after Mack, JJ1CEI and the author, 7K1NCP received SS licenses --- we attended Partech'97 in Kitakyushu-City, and demonstrated SS QSO for the first time.

Figure 1 shows the system we used in the first experiment. The PS (Protocol Server)[1] converts an IP packet into a radio packet, and the IPSM (IP Shield Machine)[2] hands it to the 2.4GHz SS data transceiver (Table 1). The antennas we used were 27 elements Yagi-beam, front gain of 21dbi.
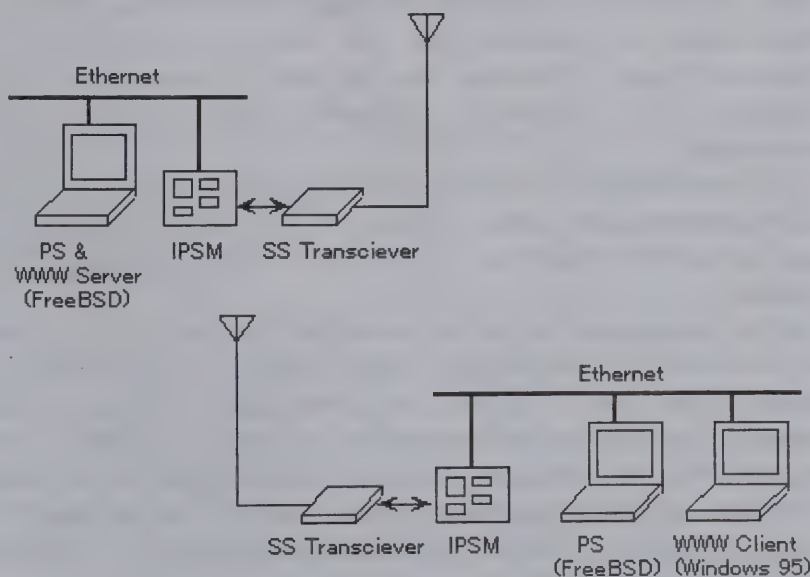

Figure 1. System configuration

Figure 2.
The first letter of Japanese old alphabet.

Table 1. The features of an SS transceiver.

Center Frequency	2446MHz
Output Power	30mW
Spreading Method	Direct Sequence
Spread Length & Code	11bits Baker, 63bits m
Chipping Rate	4.4444Mchips/sec
Occupied Band Width	8MHz
Modulation	DQPSK
Data Transfer Rate	808kbps

The first ever SS QSO in Japan was performed by JJ1CEI/4 and 7K1NCP/6, between Kyushu-Island and Honshu-Island, across the Kanmon Strait, where the distance between two stations was about 2km. Not only had we accomplished 2way SS QSO for the first time, but also this QSO has a special meaning because we used TCP/IP protocol over amateur SS packet radio.

We have measured the ping statistics, round trip time and throughput, using 'ping command' and 'web browser'. The results were 99%, 110ms and 80kbps, respectively. Figure 2 shows one of the pictures used in the experiment. The reason why we used this picture is that we wanted to share the Dr. Takayanagi's success in 1926 --- the first experiment of TV transmittance in Japan.

The second experiment in Kofu (Dec. 14, 1997)

In the first experiment, we could confirm that our system, employing TCP/IP over amateur SS packet radio, worked well with practical speed. However, the distance between two stations was only 2km, which could be achieved even with ISM band transceivers, very low EIRP (Equivalent Isotropic Radiated Power). Therefore, we decided to practice one more experiment, aiming to make a long distance QSO. Furthermore, multimedia such as voice communication was also the purpose of the experiment.

On December 14, 1997, the second experiment was done in Kofu-Valley, where we performed the 15km distance 2way QSO. The system we used in this experiment was almost the same as what we used in the first experiment. The slight difference was that we added some client PCs to each site (station).

In this experiment, we divided ourselves into two groups. One set up the base station on the top of a hill and another moved around Kofu-Valley and operated from three different points: 5, 15 and 30km from the base station. At the 5 and 15km point, the ping statistics was 100% and we tried not only web browsing but also voice communication using 'cool talk'. We could hear the voice from the other station clearly on both sides, as if they were connected to the same Ethernet.

However, at 30km point, the condition was very bad and unstable. The ping statistic shows nearly 20% at its peak and the BER (Bit Error Rate) was almost 1E-2. We changed the position slightly, but it didn't make the situation better.

The third experiment in Kofu (May. 9-10, 1998)

During wintertime, there had been much improvement in our system, though we didn't practice any field experiment. Mack tune-upped SS transceivers to make its sensitivity better. Satoshi, 7M3LCG rewrote IPSM's firmware to improve its stability. Shin, JN1JDZ implemented routing protocol into the PS. In addition, some of our members (neither Mack nor the author) obtained the SS licenses.

On May 9 and 10, 1998, we practiced the third experiment in Kofu-Valley (picture 1). In this experiment, we put the 30km distance QSO as the main purpose. The Radio-Network operation (something like a round QSO), the videoconference using CU-SeeMe and mobile communication were the purposes too.

In this experiment, we could achieve all of our purposes: We performed the 30km 2way QSO and videoconference connected three points simultaneously. Furthermore, we confirmed the following two matters: The routing table changed dynamically as we move the direction of the antenna or as the new site appeared/disappeared; We could communicate between running cars as far as they were in sight each other.
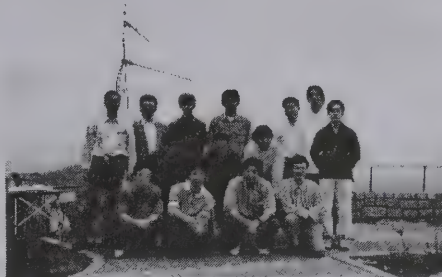
Picture 1.

Alpha/Beta tests

Since April 1998, the alpha test using our PRUG96 system has been proceeding around Meguro-Ward, Tokyo, under the support by JA1YAD/JL1ZCF, Tokyo Institute of Technology Amateur Radio Club [3]. The purposes of the alpha test are to estimate its stability, to find out its weak point and so on.

We will soon start the beta tests in many regions in Japan, such as Aichi, Fukushima, Kanagawa, Tokyo, Miyagi and Nagano. You can expect to know some outcomes of these tests in the next Partech/DCC.

References

[1] S. Kanno, not published
[2] S. Funada, IP-Shield Machine (IPSM): An Ethernet Interface for High Speed Packet Radio, this conference
[3] S. Watanabe, S. Funada, Alpha-test report of PRUG96 High speed radio link, this conference

MPT: Ministry of Posts and Telecommunications Japan
PRUG96 project: the group consists of people interested in high-speed packet links and networks
Partech: Packet Radio Technical Conference; annual meeting of amateur packet radio in Japan
Kitakyushu: a city in Fukuoka Prefecture, Kyushu-Island, Japan
Shimonoseki: a city in Yamaguchi Prefecture, Honshu-Island, Japan
Kofu: a city in Yamanashi Prefecture, central part of Japan

A New Routing Method Based on Station ID

Norito NEMOTO, JH1FBM
Packet Radio Users Group (in Japan) and TAPR member

ABSTRACT

There are two problems on routing packets by means of IP address in a seamlessly wide area radio networks: (1) conflicting IP address on routing infomations and (2) little flexibility to accomodate and isolate the differences among local network policies. To solve them , it is important to use a routing method based on the callsign of radio station. A new method is presented in this paper with an emphasis on the use of the callsign. This protocol employs Station ID (SID), which is a combination of callsign and a system number.

KEYWORDS

routing, TCP/IP, SID, SSID, NET/ROM

---

1. Routing Based on Station ID

In spite of non-unique private IP address, we are assigned highly unique call sign for each amateur radio station. This uniqueness is guaranteed by the law enforcement worldwide; an address duplicate immediately implies at most one is valid and the rest may be strongly urged to corrent the identification.

For this reason, amateur radio packet communication commonly uses a Station ID, which consists of the call sign and a system number, for example, JH1FBM-1. In this method, since the routing is performed based on SID, it is capable of handling any network protocol. The largest benefit of this method is the ability to distinguish the logical network even where some terminals of different logical networks share the same communication medium.

There are easy and better for the routing system to use the SID as physical address and exchange the IP address over SID.

2. Background

More recently, TCP/IP and NET/ROM have offered better solution. TCP/IP, in particular, has been popularly implemented in various operating systems, and attempted to apply for amateur radio communication. For the popularity and abundant application softwares already developed, it is beneficial to build our network based on IP network. However, in building networks of large scale, the capability of the protocol is currently not fully used.

When the AX.25 protocol was established, digipeating feature for packet relaying was only a temporary solution until another standard for the network layer would be accepted. Digipeating

requires for the sender to specify all the path on the way to reach the receiver; it demands too much for the sender and lacks flexibility. Therefore, it is not a satisfactory standard.

NET/ROM is 1st dynamic routing protocol based on Station ID. But, NET/ROM didn't work well as the Wide area network. Because NET/ROM capacity was limited. IP networks seem work well, but they have other problems.

3. Problems on IP Address Based Routing

Owing to rapid growth of the Internet, the IP address space is being depleted. The terminals in a private network (not directly connected to the Internet) are usually assigned private IP addresses to save the IP addresses used. Such private address can be freely used in a private network. When a terminal in a private network communicate through the Internet, the firewall of the network exchanges the packet in behalf of the terminal, to avoid the non-unique private address to appear on the Internet.

However, a new problem arises when we manage seamless wide area network, including amateur radio network, by the use of private IP addresses. This is because all the terminals exist in a same space; when two terminals have the identical IP address, the correct routing is impossible.

If we adhere to routing based on the IP address, we are left between two choices: (1) to assign Internet-valid IP address or (2) to control the use of private addresses so that unique assignment is guaranteed.

The first choice does not completely solve the problem, because there is no regulation that prohibits the use of an arbitrary IP address on amateur radio network. The second one requires a large amount of labor to manage the unique assignment of the address especially if the network is of large scale or wide area.

4. Problems on the NET/ROM.

NET/ROM worked as network protocol. But its capability was limited. A procedure of this SID based exchange system is almost same as that of NET/ROM's. We must study the reason of NET/ROM limitation to avoid making same weak point in our system.

Why the NET/ROM capability is limited.

(1) Capacity of routing table was limited. Number of table entry is much smaller than number of operating stations. This limitation is due to memory capacity of 8bit Z80 CPU.

(2) Connection type protocol's overhead is heavy. NET/ROM is designed based on AX.25. Protocol overhead is heavy and lengthy for some specific upper layer protocol. For example, TCP layer itself act as error free transfer method, it needs light lower layer to enhance system thoughput.

(3) Physical speed is limited. Using 1200bps BELL 202 modem limits address exchanging capability. Exchanging large routing table will choke network traffic.

5. How do we solve the limitation

(1) Capacity of address routing table. The size of this table for supporting a million stations will be around 16Mbytes. It is easy to get this size of memory on today's personal computer to hold the table.

(2) Connectionless lower layer. We have capability to develop original protocol stacks on open software platforms like FreeBSD or Linux. Making original connectionless protocol is not difficult. In fact, I made prototype on FreeBSD 2.2.6 and it is under stability checking.

(3) Higher speed modem. A new designed 808kbps Spread Spectrum radio is running on the PRUG96 system. This radio will serve us enough link capacity to exchange large address routing table between tens of thousands of radio stations in a few seconds. Based on a simple simulation on the PC, twenty thousands of stations will complete exchanging their routing tables each other only in three seconds.

6. Experiment

We are experimenting this protocol with PRUG96 system. And we will describe Address Resolution Protocol (ARP) , which resolve IP Address to SID of physical address. Further questions or inquiry on the latest result should be addressed by e-mail to Noriton@nemoto.com.

Reference List

[1] Norito NEMOTO / JH1FBM "FBPATH : IP over callsign exchange," 1996 PARTECH in Shodo-shima JAPAN. (Japanese paper, English abstract)

[2] CQ publishing "Packet communication handbook," CQ Publishing Corp. (Japanese book)

APRSstat
An APRS Network Analysis Tool

Richard Parry, P.E., W9IF
rparry@qualcomm.com
http://people.qualcomm.com/rparry

ABSTRACT

APRSstat monitors network traffic by connecting to an APRS telnet server and collecting network traffic statistics. The program is intended to run continuously as a background task on a UNIX/Linux system. It collects and saves network data for periods as long as a year. The data gathered by APRSstat is plotted using the companion program APRSgraph, which creates graphs as GIF images allowing them to be integrated into a web page. APRSgraph is executed as a cron job every five minutes providing near real time updates of network usage for daily, weekly, monthly, and yearly periods. Characters per minute is used to measure network traffic. A detailed example of the San Diego, California APRS network is included which is displayed on the author's home web page. A cursory explanation of the software is also included in the paper. APRSstat and APRSgraph were written in perl and use perl modules GD and Chart for creating the graph GIF images.

KEYWORDS

Packet Radio, APRS, Linux, UNIX, Networking, Perl

INTRODUCTION

The Automatic Position Reporting System[1] (APRS) is one of the most popular new facets of amateur radio today. It joins a list of amateur radio specialties that is both long and broad in scope.

Although APRS can consist of a single transmitter and a remote receiving station, what makes it particularly useful is the network. This network allows stations that would normally be limited to line-of-site distances to extend the range to large metropolitan areas. Using HF (High Frequency) communication can be extended still further and when connected to the Internet, APRS networks become global.

A network is a complex subsystem that should be monitored and analyzed to spot weaknesses and/or areas that need to be improved. Commercial tools known as *sniffers* abound for monitoring various types of networks such as Ethernet networks. WinAPRS supports fairly extensive network monitoring functionality including graphs and limited statistical traffic usage. It is the purpose of APRSstat to provide additional APRS LAN traffic information. Its major strength is its ability to run continuously as a background task to collect, save, and display long term APRS network traffic. Let's look at a real life APRS LAN to understand exactly how it works and the information that it provides.

---

[1] The APRS formats are provided for use in the amateur radio service. Hams are encouraged to apply the APRS formats in the transmission of position, weather, and status packets. However, APRS is a registered trademark of Bob Bruninga who reserves the ownership of these protocols for exclusive commercial application and for all reception and plotting applications. Other software engineers desiring to include APRS protocols in their software for sale within or outside of the amateur community will require a license from him.

SAN DIEGO, CA. APRS

First we need to understand what we are measuring, so let's define what we mean by APRS network traffic. For our purposes, network traffic shall be measured as the number of characters received per minute. Note that characters, not packets, must be used to measure network traffic since APRS packets vary in length which makes comparisons of channel usage measured by packets impractical and less useful.

APRS traffic is transmitted at 1,200 baud, therefore a single bit requires 0.833 ms (1/1200) to send. Since an ASCII character is 10 bits, the time to transmit a single character is 8.33 ms. This translates to 120 characters per second or 7,200 characters per minute which represents the ideal theoretical maximum capacity of the channel[2]. Knowing the maximum channel capacity provides us with a basis for comparison and a metric that describes how close we may be to saturating the network.

The network usage statistics that APRSstat collects are displayed as four graphs: daily, weekly, monthly, and yearly. The wide range of time periods allows short and long term tracking of the network which can then be used to spot trends.

Figure 1 shows the daily network traffic for the author's private[3] telnet server in San Diego, CA. It displays the number of characters transmitted for a 24 hour period. The vertical axis displays characters per minute. For the daily graph, the sampling period is 5 minutes. This value is primarily a compromise between providing useful traffic resolution and limiting the size of the data log. Note that the most recent data point is on the left of the graph and the oldest data is on the right. As time passes, data moves from left to right and eventually off the graph.
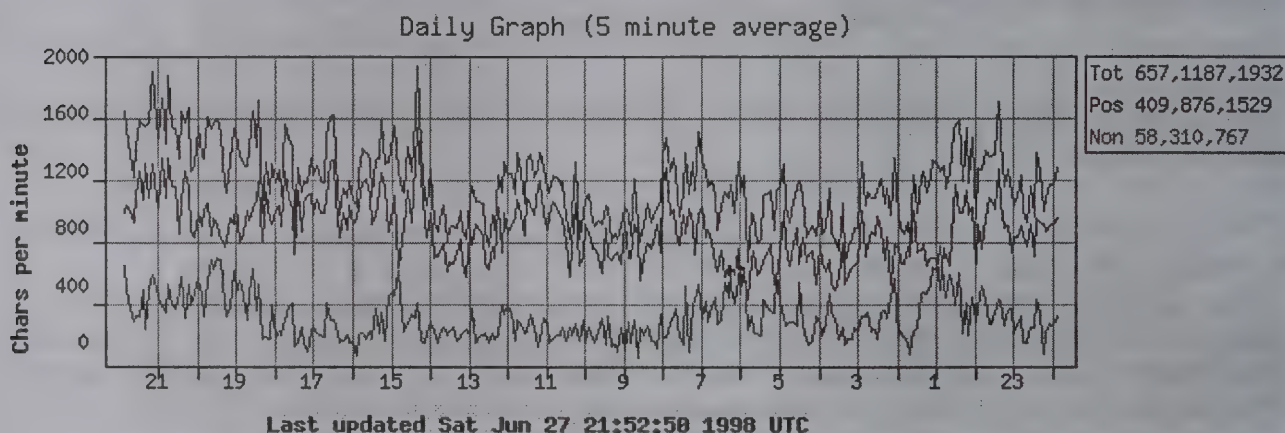
Figure 1 San Diego, Daily (24 Hour) APRS Network Traffic

To provide the most useful information, APRSstat breaks traffic into two categories: characters contained in *posit* and *non-posit* packets. A posit packet is an APRS packet that contains position information (i.e., latitude and longitude), while non-posit packets are those that do not provide position information. This is an important distinction. Although APRS was originally intended to provide

---

2   This value is an approximation since APRSstat counts payload characters only. The payload is the portion of the packet carrying user information. An AX.25 packet encapsulates the payload in a header and trailer which both require bandwidth not measured. For this reason, actual channel capacity is less then 7,200 cpm and varies with the length of the packet.

3   This private telnet server is behind a firewall and therefore cannot be reached outside the firewall. However, the results (graphs) are available at http://people.qualcomm.com/rparry/aprsstat.

position information exclusively, it now supports other types of communication. For example, it allows keyboard to keyboard communication. It also allows the transmission of weather information including: rainfall, temperature, wind direction and speed. When displayed on a web page, the graphs are color coded to help differentiate the types of traffic. Non-posit traffic is displayed in green, posit traffic in blue, and the total of the two is plotted in red.

Non-posit traffic is the lowest data set on the graph and therefore represents the least amount of APRS network traffic. Non-posit characters vary from a low of 58 to a high of 767 characters per minute (cpm) with a mean value of 310. It is worthy to note that although non-posit packets represent the least amount of network traffic, they are substantial, representing approximately 26% (310/1,187) of network traffic.

Posit characters represent the remainder of APRS network traffic. They are displayed as the middle data set on the graph. Posit characters vary from a low of 409 to a high of 1,529 with a mean of 876 cpm. Therefore posit characters account for 74% (876/1,187) of network traffic.

The data set plotted at the top of the graph is the total of both posit and non-posit packets. In the example, total characters vary from a low of 657 to a high of 1,932 with a mean of 1,187 cpm. We can now compute network traffic usage as 16.5% (1,187/7,200). For *connection* oriented protocols that support collision and/or corrupted packet detection, the network would be considered lightly loaded. Unfortunately, an APRS network must remain lightly loaded since it uses a *connectionless* protocol consisting of UI (Unnumbered Information) frames. This is just another way of saying that if a packet is dropped, there is no means to detect the loss and therefore no means to retrieve it. To drive home the point, assume an APRS network is 50% loaded. Under these circumstances a transmitted packet has a 50% probability of colliding with another packet[4] making the network almost unusable.

Figure 2 shows weekly APRS traffic for the same San Diego, CA. network. The same information is plotted, but for a longer period. This allows one to spot trends that may occur weekly. For example, there might be more traffic on weekends than during the week or more traffic during the daytime hours than at other times. The sampling period is 30 minutes for the weekly graph. An analysis shows that non-posit traffic accounts for 25% of the total network traffic. Posit characters account for the remainder, or 75%. The most important metric is total network usage which in this case is 16.67%.
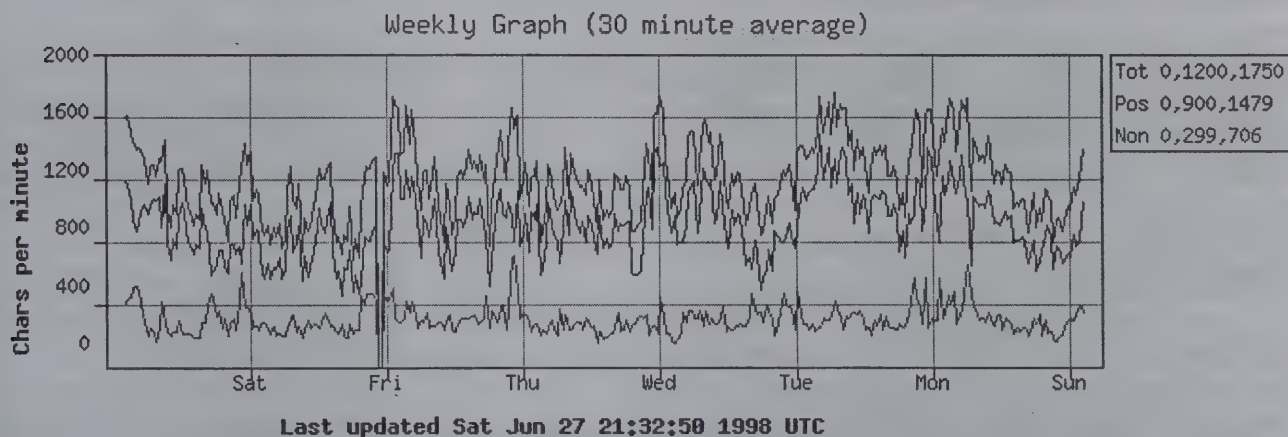
Figure 2 San Diego, Weekly (7 Day) APRS Network Traffic

---

[4] This is not completely true since the length of the packet will also determine success or failure, however the statement is sufficiently accurate for our needs.

It is important to note that the percentage of network usage by posit and non-posit traffic remains virtually constant between daily and weekly periods. This should not be a surprise since the only difference between daily and weekly data is the sampling period. Computing traffic usage for monthly and yearly periods continues to show that non-posit traffic accounts for approximately 25% of all network traffic and posit packets for 75%.

Figure 3 depicts monthly traffic. Posit and non-posit traffic for each day of the week is plotted. The sampling period is extended to 2 hours. The monthly graph allows one to spot trends that occur on a weekly basis.
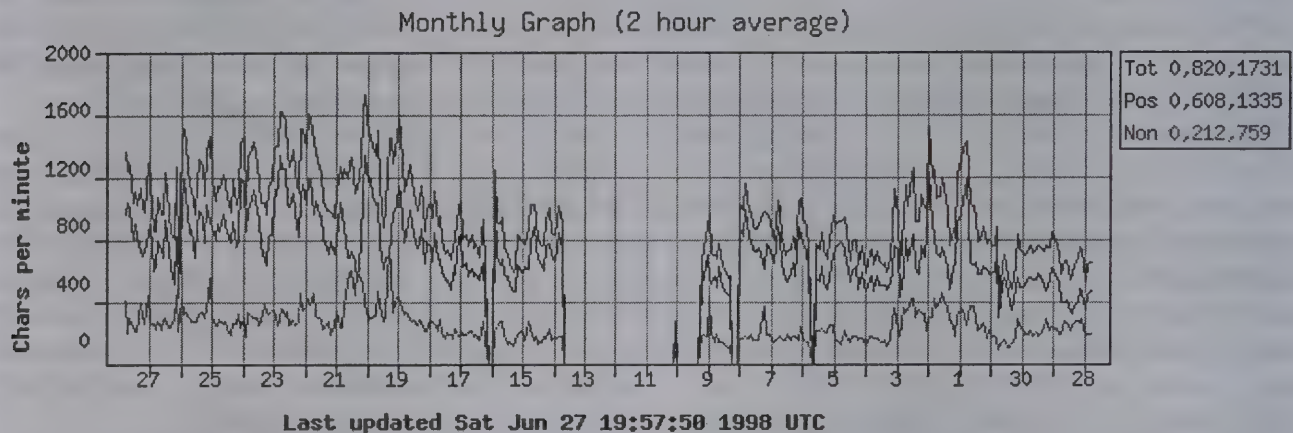
Figure 3 San Diego, Monthly (31 day) APRS Network Traffic

Figure 4 provides network traffic usage for a year. As previously mentioned, to limit the size of the log, the sample rate is extended to 24 hours. The graph begins in March since that is when APRSstat was completed and first came on line. It has been running continuously with brief periods of downtime, such as the loss of data from June 9 to 14, shown in both Figure 3 and 4. The yearly graph allows spotting long term trends that occur monthly.
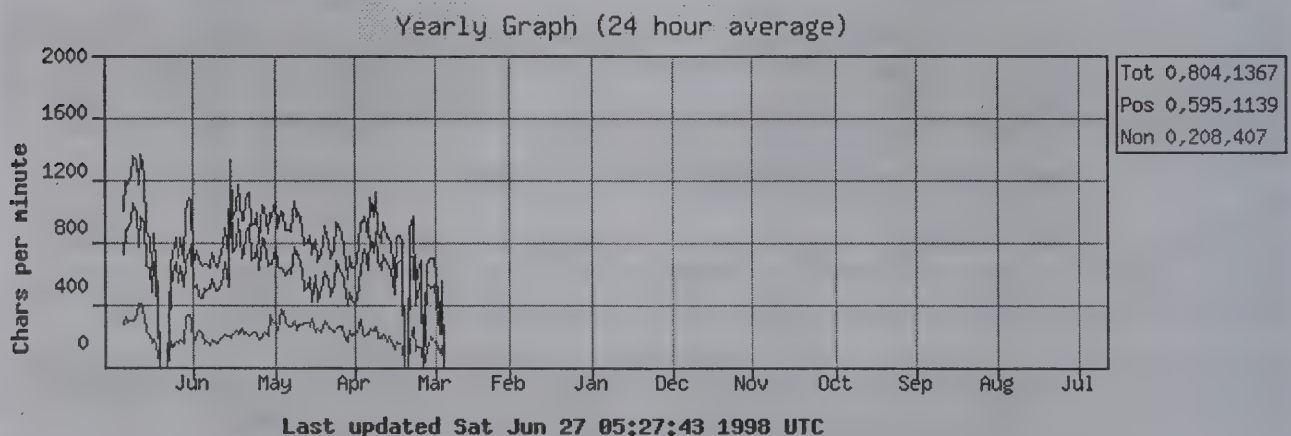
Figure 4 San Diego, Yearly (12 Months) APRS Network Traffic

WHAT DOES IT ALL MEAN

The purpose of APRSstat is to provide data in a form that is easily analyzed, and ultimately to draw useful conclusions. An examination of the graphs allows us to draw the following conclusions for the San Diego network:

1. Total channel usage is approximately 16%. It is measured by using the mean cpm and dividing by the total theoretical maximum capacity of the channel.
2. Non-posit packets represent a significant portion of APRS traffic, approximately 25%. The remaining 75% of traffic on the network comes from packets with position information.
3. It is interesting to note that daily network traffic is relatively constant. One might think that network traffic would vary with the time of day. For example, more traffic during daytime hours and less at night. However, when one realizes that most traffic contains information that is automatically transmitted (no human intervention) the conclusion makes sense. For example, position beacons are transmitted at 30 minute intervals both day and night. The same is true for packets carrying weather information. Only mobile tracker beacons and keyboard to keyboard packets are more prevalent during daytime hours. These packets are initiated manually, and therefore account for a minority of the network traffic.
4. Weekly and Monthly traffic is also relatively constant for the same reasons described above. For example, as the weekly and monthly graphs show, there is no significant increase in network traffic on weekends.
5. There is insufficient data for yearly traffic analysis since we do not have a full year's worth of data. However, with nearly 4 months of data, network traffic appears to be relatively constant. There are some monthly variations but no clear pattern is discernible. As APRS networks continue to grow, the predicted pattern should show a steady rise.

Similar results are expected for other metropolitan areas.

APRSSTAT INTERNALS

APRSstat is one of two programs that make up the system to collect and display APRS network traffic usage. It is a perl program[5] that runs as a background task under Linux, but any UNIX system should work. As we shall see in the next section, APRSgraph is a separate task that also runs in the background.

APRSstat begins with a telnet connection to an APRS server. In the examples shown in this paper, the connection is to the author's San Diego, CA, APRS telnet server. Once the connection is established, the program starts receiving APRS packets and counts the number of characters contained within the packet. It then determines the type of packet (posit or non-posit) and increments separate posit and non-posit character counters. At five minute intervals, posit and non-posit counters are time and date stamped and saved to a log file. The counters are reset to zero and the process begins again.

The log file consists of four mini-logs, one for each time period (e.g., day, week, month, and year). The first section of the log file contains the daily log data (5 minute intervals), the second section holds the weekly data (30 minute intervals), the third section contains monthly data (2 hour intervals), and the last section of the log retains the yearly mini-log (24 hour interval). The reason for the different resolutions is primarily a compromise to limit the file size.

A status log file is also provided and updated as necessary to log salient events such as when the process began and the loss of a connection to the server.

---

[5]   A perl library to be more exact.

APRSGRAPH INTERNALS

APRSgraph is a separate program and runs independently of APRSstat. While APRSstat is collecting network statistics and storing it in a log, APRSgraph is asynchronously reading the log and creating graphs on the fly. APRSgraph is executed as a cron job[6] every five minutes resulting in new GIF images which provides near real time display of network traffic.

When APRSgraph is executed, it opens the log file created by APRSstat and extracts the daily, weekly, monthly, and yearly mini-log data. Using perl modules GD.pm and Chart.pm, four GIF images are created. A web page is then used to display the information.

CONCLUSION

Networks provide the ability for two or more entities to communicate. While the entities themselves may be complex (a computer), the network can be equally complex. Understanding how the network operates is important to insure reliable communication. APRSstat was developed to provide a means to monitor APRS network traffic and ultimately to understand its behavior to allow it to be improved.

The examples provided in this paper were limited to a single APRS server located in San Diego, CA. Limited connections to other servers in metropolitan areas with significant APRS traffic show similar trends and conclusions as those outlined in this paper. It is the hope of the author to continue monitoring APRS networks as APRS activity continues to grow among amateur radio operators.

ACKNOWLEDGMENT

Thanks to Bob Bruninga, WA4APR; for permission to use the APRS trademark.

RESOURCES

1. Bruninga, Bob, "Automatic Packet Reporting System (APRS)," *73*, December 1996, pp. 10-19.
2. Dimse, Steve, "javAPRS: Implementation of the APRS Protocols in Java," *ARRL and TAPR 15th Digital Communications Conference Proceedings*, Seattle, Washington, September 1996, pp. 9-14.
3. Parry, Richard, "Position Reporting with APRS," *QST*, June 1997, pp 60-63.
4. Parry, Richard, "APRS Network Guidelines," *73*, October 1997, pp 19-20.
5. Parry, Richard, "Position Reporting: The Global Positioning System and Linux," *Linux Journal*, July 1998, pp 46-49.
6. Parry, Richard, "XNET: A Graphical Look At Amateur Radio Packet Networks," *ARRL and TAPR 15th Digital Communications Conference Proceedings*, Seattle, Washington, September 1996, pp. 64-75.
7. Parry, Richard, "perlAPRS," *ARRL and TAPR 16th Digital Communications Conference Proceedings*, Baltimore, Maryland, October 1997, pp. 141-148.
8. Parry, Richard, "Graphing on the Fly: Generating Real Time Graphs for the Web," available from the author's web page at http://people.qualcomm.com/rparry/papers.

---

[6] A cron job is a computer task that is executed on a UNIX system at specified intervals.

QUALITY ELECTRONIC MAP DISPLAYS FOR APRS MOTOR VEHICLE NAVIGATION

J. Bruce Prior, N7RR
400 Detwiller Lane
Bellevue, WA 98004
n7rr@hotmail.com   telephone 425.454.4426

> "What place would you advise me to visit now?" he asked.
> "The planet Earth," replied the Geographer. "It has a good reputation."
> -- from The Little Prince, by Antoine de Saint-Exupéry, New York, 1943: Harcourt Brace Jovanovich, Publishers

Monitoring Motion and Improving Safety

GPS receivers calculate speed and direction of travel as well as location. Coupled with APRS systems, that motion information opens exciting new applications. Changing traffic conditions can be monitored. Drivers can be alerted about speed zone changes and special traffic regulations about lane utilization and turning at intersections. Even though the electronic map system can and should contain large amounts of data, a properly-designed display system should not distract the driver with so much detail that it becomes a traffic hazard.

Problems with Paper Maps

Large folded paper maps are inconvenient to use at the best of times. In a motor vehicle they are a real traffic hazard. Even maps which are conveniently packaged as a page in an atlas must include much more information than a driver needs to solve any given navigational problem. The golden nugget which the driver needs is lost in a river of irrelevant ink. (See Tufte 1983.)

New Kinds of Maps

The advent of electronic mapping allows us to think of maps in new ways. Paradoxically, electronic maps can be much more complex and data-rich while they display only a minimal amount of information at any one time. When we change from paper maps to electronic displays, we can thereby cross a threshold of radically-improved legibility. The placement of the display is critical.

Integrated Visual Display

The wave of the future is not an external screen mounted precariously off on the passenger side. Such an arrangement takes the driver's attention off the road and interferes with important items like gear shifts, passenger-side airbags and two-meter rigs! Our visual displays should be front and center, fully integrated with fuel, oil pressure and temperature gauges, speedometer, odometer and tachometer. Incidentally, those functions need not be displayed unless they are actually needed. Does the screen need to be cluttered with a fuel gauge when you filled the tank ten minutes ago? Do we really care how fast we are going when stuck in bumper-to-bumper traffic? The speedometer reading normally becomes important only when we approach the legal speed limit. Our navigation system will know where we are and what the speed limit is. After we have set our cruise control, why take up space on the display with speed information?

A well-designed integrated visual display should automatically change the information it displays according to different situations, but the driver should also be able to call up information at will using simple control panel commands.

The best model for future motor vehicle visual displays is the kind used in state-of-art commercial aircraft. The flight deck of a Boeing 777 has five 8-inch square color liquid crystal displays. Two are duplicated and are used by each pilot. One more LCD screen sits half way between both pilots as a shared resource. In addition, a heads-up display uses combiner glass with a sandwiched layer which serves as a mirror for a particular green wavelength. Additional information is projected at that wavelength and focussed at infinity, so the pilots can simultaneously see out the windscreen and capture the green readout. (Craig 1998)

Map Data Sources

Currently most vehicle navigation software used in the United States is based on a detailed, but conceptually primitive, mapping system called TIGER, which gives block-face street data compiled for the US Census Bureau. A "block face" is one side of a street between two intersections or between an intersection and the end of a dead end street. TIGER is an acronym for **t**opologically **i**ntegrated **g**eographic **e**ncoding and **r**eferencing system. Maps based on TIGER are reasonably accurate topologically, and provide an excellent source of place name information, including streets and roads (Clarke 1997: 119-120). With some exceptions, streets which are shown connected by the system are actually connected on the ground. Since the block face information includes a range of numbers, the location of any particular address can be designated within a block of its location relative to the nearest intersection. When viewed at large scale, say, 1:10 000 or larger, the geographical inaccuracies of the TIGER data become evident. Curved streets look straight and distances between any two points on the map are not reliable.

In the United States, geographically-accurate data can be obtained best from the United States Geological Survey (USGS). Recently the USGS has made its 1:24 000 and 1:25 000 topographical maps available on CD-ROMs. The CD-ROMs cost considerably less than the equivalent maps in paper form. Some private companies have begun to release software based on USGS topographical data. The biggest problem with USGS data is that place names, including streets and roads, are poorly represented.

To develop a new kind of electronic map, therefore, we need to begin by integrating the street address place names data of TIGER with the superior geometric accuracy and elevation data from the USGS mapping system. Then we need to enhance the resulting data with information which is important for motor vehicle navigation. We have to collect data on speed limits, on one-way streets, on roadway dimensions and lane designations.

Managing Traffic

Monitoring the motion of a motor vehicle can go considerably beyond tracking its coordinates and tracing that motion on a map. Motion monitoring can be used to help manage traffic congestion. If a GPS

receiver is linked with a sophisticated geographic information system (GIS) through APRS, the system in the vehicle can telemeter traffic congestion to other vehicles with no intervention by the drivers when it determines that the vehicle is traveling on an arterial at a speed which is markedly slower than normal.

When properly implemented, the system will be able to tell, for example, that regular southbound lanes on Chicago's Kennedy Expressway are crawling at about 10 MPH just north of the Eisenhower Expressway, but that the express lanes are operating normally. On a snowy morning in downtown Boston, drivers will be warned to avoid using steep Joy Street to climb Beacon Hill. Oregon drivers will learn that part of US Highway 26 has been closed due to mudslide hazard associated with increased volcanic activity on Mount Hood. The navigation system in a pizza delivery car in Fort Worth which is already exceeding the city speed limit will warn its harried driver with a loud audio alarm and visual display that she is fast approaching an active 20 MPH school zone.

Inaccuracy Problems

The system envisioned here presupposes accuracies of both map and navigational measurements which are better than those generally available today. First, the geocoding of the electronic maps should be accurate with a resolution of one meter or less. Such accuracy will allow positive identification of a particular traffic lane on any road. County and municipal engineering staffs generally maintain survey information about road networks under their jurisdictions which are accurate within a few inches (Carpenter 1998). A reasonable goal would be to attain accuracy resolutions of about one decimeter or about four inches for road networks. We need to gather such geographic information for our databases and keep it updated.

Second, navigational accuracy needs to be higher than those achieved by standard civilian GPS receivers. Typical GPS receivers such as the older Magellan Trailblazer XL or the more recent Garmin GPS 12 XL display waypoint definitions using the Universal Transverse Mercator coordinates to a resolution of one meter. Basic GPS should normally be accurate within about 10 meters horizontally and 13 meters vertically, but the US Department of Defense (DOD) has imposed random accuracy degradation, called "selective availability," on civilian users of GPS, resulting in accuracies within about 40 meters horizontally and 50

meters vertically.[1]

Better accuracy can be achieved several ways. First, the DOD has announced that selective availability will be removed as soon as technology is developed to foil enemy use of the system in combat situations. Second, receivers can integrate signals from both the American GPS and the Russian GLONASS satellites (Daly and Misra 1995). Third, internal monitoring of vehicle tire motion and revolutions can supplement GPS navigation with dead-reckoning (French 1995: 284-286). Fourth, GPS calculations can be considerably improved through the use of differential GPS transmitters (Dahl 1993: 157-162), whose precisely known locations can be used to calculate errors in GPS transmissions caused by selective availability and inherent system errors such as variations in the ionosphere and signal multipathing (B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins 1997: 126-130). Finally, nearby fixed "pseudolites" (Elrod and Dierendonck 1995), or ground-based transmitters which mimic operational navigational satellites can improve accuracies when joining the chorus of the signals transmitted by the GPS satellite constellation.

Map Legibility

Legibility of maps is a cartographer's highest priority. Coupled with the motion-detecting capability of a GPS system, electronic maps need to display only a tiny portion of the information which they contain. A driver navigating the streets of Berkeley electronically can have access to much more detailed information than any paper map can provide without being burdened by unneeded clutter. For example, a Bay Area system can and should include topographic information. Drivers should not be expected to interpret the meaning of elevation contour lines. Rather, those contours can be left undisplayed, but the system can use the information to calculate the grade percentage of a particular road segment.

Place Name Precedence

Take a look at a globe with place names on it. You will very likely be able to find Barrow, Alaska. Now try to find Evanston, Illinois on a globe. According to the last census, Evanston had more than twenty-one times the population of Barrow, but you will not find it on any world

---

[1] These figures represent one standard deviation. They are actually 10.2 m horizontally and 12.8 m vertically without selective availability and 41.1 m horizontally and 51.4 m vertically with selective availability. (Parkinson 1994: 481-483)

map. Barrow holds the distinction of being the northernmost town in the USA, surrounded by the usually-frozen Chukchi Sea to the west, the Beaufort Sea not very far away to the east, and a lot of tundra to the south. Evanston also sits astride a big body of water, but on a map of the world, there is no room to print "Evanston." The space is probably already occupied by the first "C" in Chicago or the "L" in Lake Michigan.

Place names are important. When they are required on a given map, professional cartographers give them top precedence. It is acceptable to break a road or a river or a grid line or a boundary to make a place name visible, but never the other way around. Applying this classic cartographic principle on a moving electronic map can be tricky. Place names need to move as the map rotates so they are always legible. They must never overlap one another.

Maps produced by the National Geographic Society are good models for the use and placement of place names. Upper and lower case type is the rule. Only the largest regional features sport all capital letters. The reason is that upper and lower case letters, especially when they have to be small, are easier to read. When was the last time you read a novel printed completely in capital letters?

Speed-Driven Scaling

When planning electronic maps to be used in motor vehicles, we need to consider carefully which place names are needed for a given navigational job. For a vehicle traveling at highway speeds, only place name information which is relevant to upcoming intersections or exits needs to be included. As the vehicle accelerates, the scale of the displayed map should decrease, covering a wider area, and it should show fewer place names. Conversely, as the vehicle slows down upon entering a residential or business district, the map scale should increase, showing a smaller area with more detail, including more place names.

Color Contrast Legibility

Some years ago, AEA built the MM-3 keyer, called The MorseMachine. The keyer was a well-engineered product which unfortunately is no longer in production. The designers put a command summary chart on the outside of The MorseMachine which includes good and bad color contrasts. The two worst are black type on a red background and black on green. Black on light blue is a bit better. Black on white is much

better and black type on a yellow background gives the very best color contrast legibility. Highway caution signs are painted black on yellow for a very good reason.

When we design a visual display system for mobile electronic maps, we can and should include color contrasts. City streets could be color-coded by a smart mapping system based on whether the vehicle can turn onto it from a particular approach. The display in a vehicle heading north and approaching a one-way street set up for easterly travel might show the eastern street segment in yellow to show that a right turn there is possible, while the western street segment could be colored gray to indicate that a left turn there is not acceptable. Similarly, a very narrow crossing street might be colored gray in both directions for the system in a pickup truck which is pulling a long mobile home, indicating that the intersection is too tight for a wide-turning vehicle.

Less is More

To summarize, in order to produce quality electronic map displays for APRS motor vehicle navigation, we need to accomplish these steps:

-- Take advantage of the motion-monitoring capabilities of GPS to share relevant traffic conditions automatically with others in an APRS network.

-- Replace peripheral displays with comprehensive central displays.

-- Increase the accuracy of navigation hardware to one meter or less.

-- Increase the accuracy of geographic information databases to one decimeter or less.

-- Make displays more legible by using upper and lower case place names and carefully-designed color coding.

-- Supplement critical visual warnings with audio alarms.

-- Restrict to a bare minimum the amount of information displayed to that which is absolutely necessary for navigation.

BIBLIOGRAPHY

Beth Carpenter 1998
       Personal Information, Public Works Department, City of Redmond, Washington

Keith C. Clarke 1997
       Getting Started with Geographic Information Systems, New Jersey: Prentice Hall

Ray Craig 1998
       Personal Communication, Test Pilot, The Boeing Company, Renton, Washington

Bonnie Dahl 1993
       The Users Guide to GPS: The Global Positioning System, Chicago: Richardsons'
Marine Publishing

B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins 1997
       Global Positioning System: Theory and Practice, New York and Wien: Springer-
Verlag

Bradford W. Parkinson and James J. Spilker, Jr. 1996
       Global Positioning System: Theory and Applications, Volumes I and II, Wash-
ington: American Institute of Aeronautics and Astronautics, Inc.

       Peter Daly and Pratap N. Misra 1995
              "GPS and Global Navigation Satellite System
              (GLONASS)," in Parkinson and Spilker 1996: II:243-272

       Bryant D. Elrod and A. J. Van Dierendonck 1995
              "Pseudolites," in Parkinson and Spilker 1996: II:51-79

       Robert L. French 1995
              "Land Vehicle Navigation and Tracking," in Parkinson and Spilker 1996:
       II: 275-301

       Bradford W. Parkinson 1994
              "GPS Error Analysis," in Parkinson and Spilker 1996: I:469-483

Edward R. Tufte 1983
       The Visual Display of Quantitative Information, Cheshire, Connecticut: Graph-
ics Press

An Inexpensive PC-Modem for 76.8kBit/s User Access

Thomas Sailer, HB9JNX/AE4WA,

and

Johannes Kneip, DG3RBU

August 14, 1998

Abstract

This article describes a simple and inexpensive modem intended to link end users at 76.8kBit/s to the high speed backbone network. The modem can be connected to standard PC's using the Enhanced Parallel Port (EPP) interface.

1 Introduction

The beginning of this project dates back about two years ago. The emergence of a high speed backbone led to the question of how to inexpensively link end users at appropriate speeds to the backbone.

One of the primary design goals was simplicity. We therefore decided to stick with FM and the well known G3RUH modulation format and just scale the technology up by a decade. This led to the allocation of a wideband duplex channel (200kHz per direction) in the 70cm band and the development of an appropriate transceiver [9].

Now the question was how to connect the transceiver to a computer. Most TNC's currently in use are hard pressed to operate at 9.6kBit/s, and therefore are inappropriate for data rates around 100kBit/s. Also, most TNC's connect to the host computer via the serial interface. With radio bit rates approaching the maximum bit rate of the serial interface of todays PC's, this interface becomes the bottleneck. While there exist TNC designs capable of doing 100kBit/s, they were considered too expensive (>$500).

Today, Packet Radio operators use graphical operating systems on their computers and want to use web browser technology also on amateur radio. Most popular amateur radio BBS software already has increasingly popular HTML interfaces to their message base, and HTTP servers are mushrooming also in Amateur Radio installations. These applications made TCP/IP popular, in fact this converted several well known TCP/IP adversaries on the amateur bands to TCP/IP users!

With TCP/IP a requirement, TNC's add little value. Most TNC's need to be switched into a dumb packet IO mode using protocols like 6PACK or KISS, requiring the host CPU to implement AX.25.

2   Design Considerations

As seen in the previous paragraph, we can save cost by implementing AX.25 in the host CPU. Todays PC's can do this with negligible overhead. Now the question is whether HDLC should be done in software or hardware. Measurements on several popular processors as well as older ones (table 1[1]) show that contemporary PC processors can do HDLC encoding and decoding up to a few hundred kBit/s with little overhead. It was therefore decided to do the HDLC encoding and decoding in software, making the hardware adapter even less complex.

CPU	CPU clock MHz	Bogomips	Encoder MBit/s	Decoder MBit/s
Intel 486DX2	66	33	4.31	3.20
Intel Pentium	75	30	6.24	5.64
Intel Pentium	100	40	9.37	7.48
AMD K5	100	200	15.00	12.69
Cyrix 6x86MX	166	166	19.51	15.53
UltraSparc 1	166		25.75	19.16
AMD K6	200(?)	465	24.73	19.05

Table 1: Software HDLC encoder and decoder throughput

The next decision was what interface to use. The interface had to fast enough including headroom for expansion (excludes RS232 serial ports), widely available (excludes USB), and simple to use (excludes USB, ethernet). The remaining interface was the Enhanced Parallel Port, which is available on virtually any computer for several years already and which was standardized by IEEE [2].

We decided to use an already existing modem design [7]. The adapter to be designed thus had the following requirements:

- EPP interface

- Synchronous serial interface according to [7] to connect to existing modems [7, 6]

- Provide elastic buffering to allow block-wise processing of the data by the host CPU

3   The Initial Design

A search for suitable components ended at IDT's FIFO circuits 72131 and 72132 [4], which ideally suited the above requirements, since they additionally also contain a parallel to serial or serial to parallel converter

---

[1]Benchmarking was done using "optimized" C language, similar to what is in the Linux driver for this adapter, see for example linux-2.1.108:drivers/net/hamradio/baycom_epp.c

shift register respectively. Adding some glue logic and drivers to provide high drive capabilities on the EPP signals completed the initial design, which was published in [5].

Unfortunately, these IDT IC's are quite expensive, and are getting more expensive, contrary to the general trend in microelectronics. The external modem adds to the cost, too. We therefore considered a redesign.
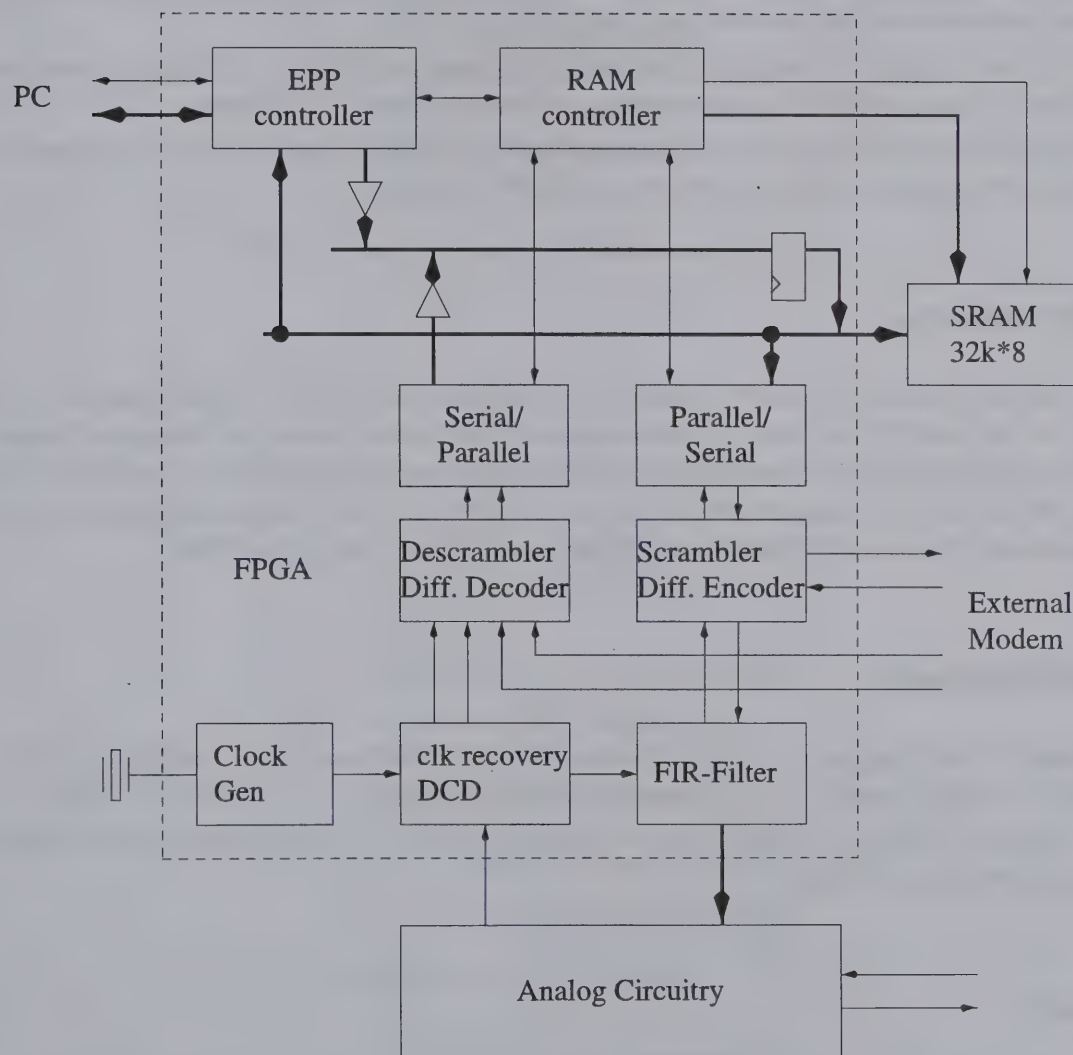
4 The Current Design

Figure 1: Block diagram of the circuit

An alternative to using dedicated FIFO IC's is to use standard SRAM and add appropriate control logic. Figure 1 shows a block diagram of the circuit of the redesigned adapter. The thick lines represent data paths, while the thin lines show control signals.

A system clock frequency in the range of 10–20 MHz seems adequate for bit rates in the 100kBit/s range and for the maximum transfer rate of the EPP port in the range 1–2 MByte/s. The relatively moderate clock frequency and the size of the logic required to implement the modem adapter suggest the use of a FPGA (Field Programmable Gate Array) to implement it.

Unlike other programmable logic families, FPGA's store their configuration in static RAM cells, which loose their information at power down. At power up, they need to reload their configuration data, which may originate from a special serial PROM, a standard EPROM, or directly from the PC. The latter was chosen since it is more flexible, cheaper and easier to develop. Also, modem options can be realised by patching the configuration data prior to download to the FPGA.

The configuration data is downloaded to the FPGA using the IEEE 1147 JTAG ("Joint Test Access Group") protocol. The protocol also allows testing of various parts of the adapter.

The building blocks of the modem will be described in more detail in the following subsections. Figure 6 shows the circuit diagram of the modem.

4.1   EPP Controller

The EPP controller handles the EPP protocol [2, 3] together with the PC. Data read and write cycles directly access the data FIFOs, while address read and write cycles access the status and control register respectively. Standard PC EPP controllers emit EPP address and data cycles on IO accesses to their base address +3 (0x37b for LPT1) and +4 (0x37c for LPT1). EPP controller programming details can be found in their data sheets [10, 11, 12]. Tables 2 and 3 list the meanings of the register bits.

4.2   RAM Controller

Figure 2 shows a block diagram of the RAM controller. The RAM controller coordinates RAM accesses, keeps address and byte counters, and generates the control signals for the static RAM access.

Figure 3 shows a read cycle, while Figure 4 shows a write cycle. The relatively slow cycles allow the use of inexpensive standard RAMs.

4.3   SRAM

The SRAM implements the data storage for the FIFOs. 32kBytes are huge for the application, but smaller RAM's are either more expensive or no longer available at all.

Bit	Purpose
7	0: DCD an
5:4	transmitter FIFO
	00 $\geq$ 0 bytes free
	01 $\geq$ 255 bytes free
	10 $\geq$ 1792 bytes free
	11 $\geq$ 1023 bytes free
3	PTT (transmitter FIFO not empty)
2:1	receiver FIFO
	00 $\geq$ 1793 bytes stored
	01 $\geq$ 1025 bytes stored
	10 $\geq$ 0 bytes stored
	11 $\geq$ 256 bytes stored
0	receiver FIFO not empty

Table 2: status register (EPP address read cycles)

Bit	Purpose
7	LED: STA
6	LED: CON
5	external modem connector: RESET
4	transmitter FIFO enable
3	receiver FIFO Enable
2:0	interrupt rate or FIFO status
	000 interrupts off
	001 read receiver FIFO count
	010 read transmitter FIFO count

Table 3: control register (EPP address write cycles)
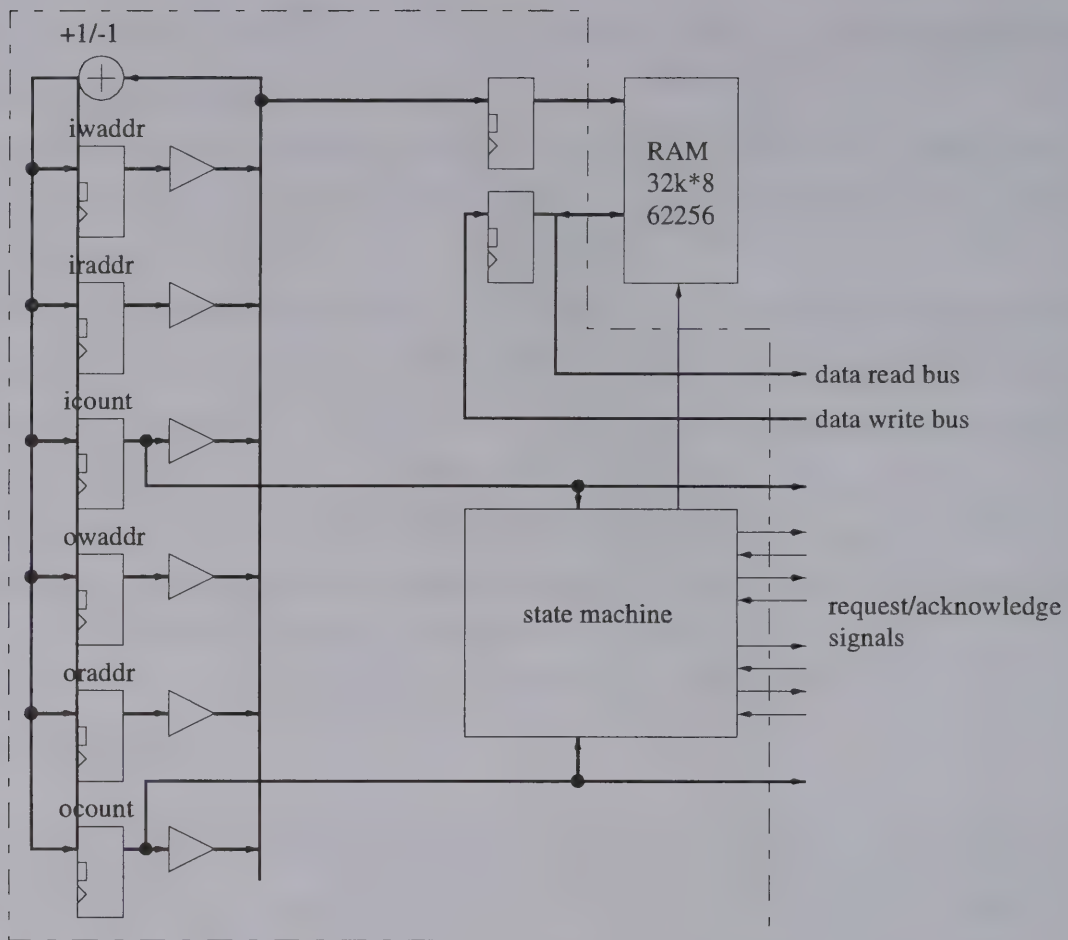
Figure 2: RAM controller

4.4   Serial to parallel and parallel to serial converter

The serial/parallel converters are doubly buffered, i.e. they contain an additional storage register besides the shift register. These blocks also contain the control logic to request emptying/filling of the storage register by the RAM controller and to transfer data between storage and shift register.

4.5   Scrambler/Descrambler

These blocks contain a G3RUH compatible scrambler/descrambler and a differential encoder/decoder. Both blocks can be bypassed independently to accomodate external modems which usually already contain one or the other functionality. These blocks also contain the switch between the internal and the external modem, as well as synchronisation circuitry for the external modem signals.
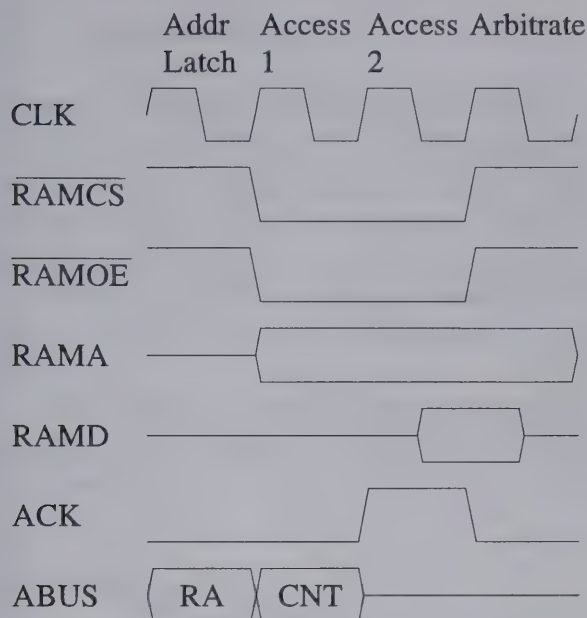
Figure 3: RAM read cycle
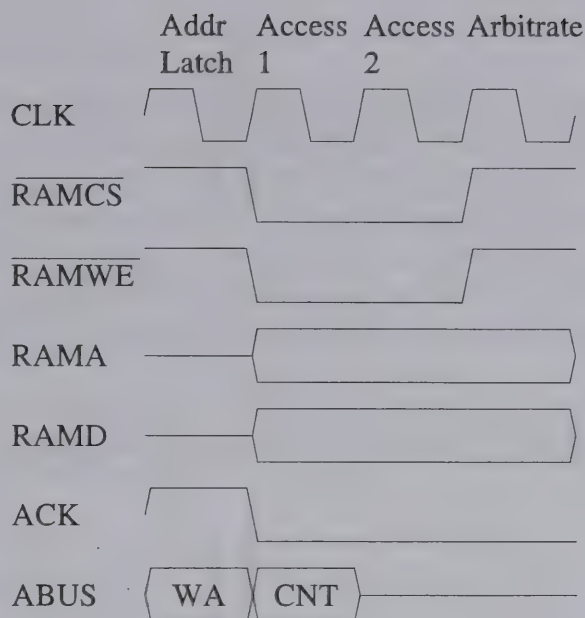
Figure 4: RAM write cycle

4.6 Receiver clock regeneration and DCD logic

The clock recovery block extracts the receive clock from the input data signal using a digitally implemented PLL. The input signal is oversampled 16 times. The DCD signal is generated from the phase values of input signal edges.

4.7 FIR Filter

Just like DF9IC [7] and the G3RUH designs, this modem too uses a four fold oversampling FIR filter of length 32. The oversampling simplifies the analog transmitter filter. Unlike the aforementioned modems, this modem does not use a table with precomputed filter values, instead it calculates the filter output for each cycle from the input data and the filter coefficients. This architecture suits the FPGA better and an external EPROM with the filter table can be omitted.

Since the modem uses an internal clock which is 16 times faster than the transmit clock, but the filter is supposed to be four fold oversampled, only 4 clocks remain to calculate the filter value. Fortunately, out of the 32 input values, 24 are zero thanks to the oversampling, and the other 8 are either $+1$ or -1. Therefore, one needs to perform only 8 adds to implement the filter. Since there are only 4 cycles, the filter is split into two halves with a dedicated final adder adding the outputs of both halves. Figure 5 shows the circuit.

Figure 5: Transmitter FIR filter

4.8   Clock generator

The clock generator divides the frequency of the external oscillator by a variable divisor in the range 1–1024 and feeds the internal modem. The maximum clock frequency of the FPGA is 20 MHz.

4.9   Analog Circuitry

The analog circuitry was mostly taken from an existing modem [6]. The frequency determining components have been scaled to the intended bit rate (76.8kBit/s) and the DAC has been replaced by a resistor array for cost reasons.

5 Conclusion

Together with the transceiver in [9], this adapter provides cost effective high speed access to the fast backbone currently being built in central europe. While there are more bandwidth efficient solutions, they are usually considerably more expensive. The design fits onto a two layer PCB in half eurocard format (10cm × 8cm) using standard through hole components. The kit can therefore be built by less experienced amateurs. Kits and ready-made devices should be available by the time of publishing from Baycom [8].

Since the redesigned adapter is mostly compatible to the earlier design, there are already drivers available for major operating systems, such as a FlexNet driver for DOS/Windows95/Windows98 and a Linux driver.

The design is quite flexible. Although it has an internal modem, it can still bypass the internal one to accomodate an external one should the need arise. It also provides enough headroom to increase its bit rate if necessary.

6 Outlook

The prototype has been successfully tested running at 1MBit/s. The CPU overhead however was considerable, 25% measured using a 100 MHz Pentium computer. The bigger part of the overhead comes from the IO.

This could likely be changed by using ECP instead of EPP. Enhanced Capabilities Port (ECP) is an alternative parallel port protocol initially designed by Microsoft, but now also standardized by IEEE [2]. All parallel port implementations these days also support ECP. In ECP mode the controllers implement a FIFO to decouple CPU and parallel port, and they may use DMA to transfer the data between FIFO and main memory. Data transfer could therefore be offloaded from the CPU to the DMA controller.

The downside is that the ECP protocol is considerably more complex than the EPP protocol, especially data transfer direction changes are expensive. Also, the Microsoft reference implementation, which just about everyone copied, has additional restrictions not inherent to the protocol.

References

[1] Xilinx Corporation *The Programmable Logic Data Book*
    http://www.xilinx.com

[2] *IEEE Standards BBS Information Technology*
    http://standards.ieee.org/catalog/it.html

[3] *Introduction to the IEEE 1284-1994 Standard*
    http://www.fapo.com/1284int.htm

[4] IDT Corporation *Specialized Memories & Modules*
    http://www.idt.com

[5] Wolf-Henning Rech, DF9IC, Johannes Kneip, DG3RBU, Gunter Jost, DK7WJ, und Thomas Sailer, HB9JNX, *Ein Modemadapter für den EPP*, 41. Weinheimer UKW-Tagung, Skriptum, 1996

[6] Johannes Kneip, DG3RBU, *Das FSK+-Modem mit Echoduplex*, Adacom Magazin 10, 1997

[7] Wolf-Henning Rech, DF9IC, *Modernes FSK-Modem – kompatibel zum Standard nach G3RUH*, Adacom Magazin 2, 1991

[8] Baycom ⬦ Hard- und Software GmbH, Bert-Brecht-Weg 28, D–30890 Barsinghausen, phone ++49 (5105) 585 050, fax ++49 (5105) 585 060, `http://www.baycom.de/`, Email: `baycom@baycom.de`

[9] Martin Liebeck, DL2ZBN, und Alexander Kurpiers, DL8AAU, *Hochgeschwindigkeits-Packet-Radio – Baugruppen für das 70cm Band* Adacom Magazin 10, 1997

[10] *Standard Microsystems Corporation (SMSC) Personal Computer Input/Output Products* `http://www.smsc.com/main/catalog/pcio.html`

[11] *National Semiconductor Product Catalog: SuperI/O* `http://www.national.com/catalog/PersonalComputing_SuperIO_5Volt.html`

[12] *Winbond Serial and Parallel Port Controller* `http://www.winbond.com.tw/produ/perso5.htm`

Figure 6: Circuit Diagram

Take the Next Step with the Next Generation Protocol

Copyright © 1998 Naoto SHIMAZAKI
<shimaz-n@pg.highway.ne.jp>
Sat Aug 15 23:37:54 JST 1998

This document describes an idea for use of IPv6 over the amateur radio. IPv6 has huge address space and it supports realtime traffic. IPv6 realize new applications. For example, managing IPv4 address is not easy. It is possible to encode our "call sign" into IPv6 address. It enables us to managing IPaddress much easier.

Table of Contents:

1. What's IPv6

IPv6 is a next generatin of IP. IP is a inter networking protocol used in all over the world. Since the IP we are using today is version four, the current version of IP often called IPv4.

1.1. What the IPv6 Has Advateges of

The most important advantage in IPv6 is its design concept. That is,

Keep Concept

IPv6 keeps good concept in IPv4. This policy is based on the fact that IPv4 has been a major player on the Internet for more than two decades. IPv6 specification is fixed by deleting groundless spec, not used spec in IPv4, keeping good spec and bringing in new idea. And, IPv6 header format was simplified and became easy to implement.

Extended address space besed on statistical estimation

The new address space is set to be large enough to connect 1000 trillion computers through one trillion networks.

The huge address space of IPv6 gives much deeply hierarchical address structure instead of only three layers on IPv4. And the huge space gives quite new address usages.

Careful Planning of address assignment

The meaning of any part of the adress space will be dicussed and agreed before practical use. Many address regions are reserved for future use. These reserved addresses are used not only when it faces the address shortage but also when new idea of address use is introduced.

1.2. Simplified Header Format

The header format of IPv6 is simplified. Here is the list of interesting changes in header fields.

version field

The very first 4bits are verion field as same as IPv4. Their contents are no longer important becouse IP version is identified by link layer protocol instead of this field.

priority value

The next 4bits indicate priority. This field is used to support realtime traffic.

flow label

Newly added field. This field is used to suport realtime traffic.

payload length

The total length field in IPv4 is replaced by pay load length field. The pay load length holds the length of "payload data" in the packet. The pay load length isn't including header's length. In othre words, subtracting header's length from total length gives payload length.

next header

One or more extention headers may follow the first header. Next header is identifier of nearest trailing extentional header.

hop limit

Same as time to live (TTL) in IPv4 but the ditails of specifications are simplified.

The fields for header length, type of service, identification, flag, flagment offset, header checksum are ommited. Considering experience in IPv4, these fields are redundant and waste in routers and rarely used efficiency.

1.3. Extention Header

IPv6 can handle variouse optional information on consystent system. Several kind of extention headers are already defined for routing, authentication and security. The option field in IPv4 is absolutly redesigned. Today, option field in IPv4 is rarely used.

1.4. Extended Address Space

The systematic use of huge address space is main concept of IPv6.

IPv6 has 128bit address space. The first 10bits are defined to categorize the address regions. They are categorized on meaning of address region. Other hand, IPv4 address is devided only by size of address region.

1.5. Realtime Support

IPv6 supports the realtime traffic by effective use of flow label and priority. The supporting realtime of IPv6 is based on "fair queuing" mechanism.

Normaly, router has only one queue par interface. The packets arrived at the router are always put into the last of queue. The packet arrived first reaves the router first. The order of packets is never changed. In fair queuing, router has two or more queue par interface. Each queue has own priority. The packets arrived at the router are put into certain queue matching priority of each packet. The router processes the queues to satisfy priority of each queue. This meens that the higher priority packets reave the router more quickly.

IPv6 router defines queue the packet to be put in by priority and flow label in packet header.

Some router partly supports realtime traffic on IPv4. However, IPv6 router gives much better support.

1.6. What Isn't IPv6

IPv6 is:

- Not a "rich" IP
- Not a "light waight" IP

Unfortunately, you would not get any interesting effects if you replace IPv4 by IPv6 on the intranet in your offece. Because IPv4 has enough ability to handle usual intranet requirements. So, why IPv6?

2. Why IPv6?

2.1.　　　IPv6 Targets Exciting Applications

The diffreence between IPv4 and IPv6 is their target application. IPv4 has been used to connect computer-based applications. Compared to this, IPv6 will connect "everything," like phones, sensors, handy tools, and others.

The realtime applications will change their network platform from IPv4 to IPv6. Because the applicatins target commonplace person, not computer mania, who like to use more stylish tools to access their applications. Much more number of tiny goods are likely to be IPv6 node becouse IPv6 has huge address space.

IPv6 realizes the mobile networking using IP handover. The IP handover is new technique to change IP address of specific node without disconnecting TCP session. It is like handover among mobile phone base stations. This handovvver feature is not provided in IPv4.

IPv6 is expected to be common inter connection technology used in every field includes home automation, industrial monitoring, mobile phone, education and much more.

2.2.　　　IPv6 Supports Realtime Traffic

the world wide web is a "cllasical" application today. More attractive voice and moving picture realtime application is drawing people.

Some important technical elements to make highspeed and flexible radio data link have been developed in amateur radio.

2.3.　　　IPv6 Can Automate the Address Management

Most important problem for us in IPv4 is the address management. The IPv4 addrsss is managed by human. It is defficult to manage the address automaticaly on IPv4. IPv6 can solves this problem. Their addresss of netowrk protocol must be unique each other. Getting unique address without centrerlize adminstaratin is difficult.

It is pretty nice idea to map specific identifier which is alrady managed and not to conflict each other into IP address by one-to-one translation. It enable us getting unique IP address without new addresss administration.

We have "call signs" of radio stations and they are unique each other among all over thte world. IPv6 has enough address space to hold our call sign after mapping. Compared to this, no enough bits are left for optimize routing if the call sign is put int IPv4 address space. Because the name space of call sign is about as large as IPv4 address space.

Also IPv6 provides something other plug and play fueature. For example, IPv6 node can completely initialize itself with getting temporaly IPv6 address automaticaly using Ethernet pysical address.

IPv6 is easy to use. IPv6 has great possibiility of various new applications. IPv6 will give you absolutly new world. So why not IPv6?

3. Address Mapping

This section describes an idea of mapping call sign to IPv6 address one-to-one.

Our call sign consists of number digit and alphabet. Single letter of the call sign is to be encoded into one six bit binary integer.

Here is the tarnslation table between letters of call sign and integer.

```
call sign letter | integer

------------------+---------
    " "       |  0
   "0"        |  1
   "1"        |  2

   ...        |  ...
   "9"        |  10
   "A"        |  11
   "B"        |  12

   ...        |  ...
   "Y"        |  35
   "Z"        |  36
 reserved     |  37

   ...        |  ...
 reserved     |  63
```

The call sign is tarnslaged into an array of six bit binaryes, reveresed, and put in IPv6 address. The reaseon of reversing alignment of the encoded call sign letter will be discussed later.

Here is an example figure of IPv6 address. The encoded call sign is "7L4FEP", which is my call sign.

```
 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   reserved for network prefix          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   reserved for network prefix          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   reserved (1)            |  "P"   |"E"|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  "E" |  "F"  |  "4"  |  "L"  |  "7"  | (2) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

(1) reserved for network prefix or call sign extention. (2) host address space

The last 4bits are used for host address. Each radio station is able to have at most 14 hosts directly connected to forein world.

The first 64bits are reserved for network prefix. For the present, some fixed bit pattern is assigned.

The bits from 64th to 87th are reserved for future use. It should be zero. If the length of call sign is extened from six letters to seven letters, the bits from 82th to 87th are used fo extended letter. If the network prefix has to be longer, the bits from 64th to 81th are used. If it has to be much longer, the bits from 82th to 87th are used. Then, these bits became unusable for call sign extention.

On other words, the encodes call sign grows tail to head and the network prefix grows head to tail. That is the reason of reversing alignment of the encoded call sign letter.

4. IPv6 Implementations and Ported Application Programs

Many of software platforms like Linux and BSD are going to support IPv6 today. Major internet application programs are already compatible with IPv6. Refer to this URL to get useful information about IPv6.

http://www.terra.net/ipv6/ <http://www.terra.net/ipv6/>

Recent version of Linux kernel already supports IPv6 and fair queueing. I confirmed it by linux-2.1.115. See this page.

http://www.kernel.org/ <http://www.kernel.org/>

ftp://ftp.kernel.org/ <ftp://ftp.kernel.org/>

There is the project to implement IPv6 stack to FreeBSD. See this page.

http://www.kame.net/ <http://www.kame.net/>

WIDE IPv6 working group.

http://www.v6.wide.ad.jp/ <http://www.v6.wide.ad.jp/>

5. Conclusion

It gives us much merit makeing amateur radio network on IPv6 with address mapping method described here.

Specific applicatin to use IPv6 in amateur radio is under development.

We need some skills to use IPv6 now becouse software platform like Linux is no stable when used with IPv6 today. But soon it will be easier on amateur radio to use IPv6. Net work newcomer will enjoy data communication on amateur radio.

----- here ----- here ----- here ----- here -----

Naoto SHIMAZAKI
shimaz-n@pg.highway.ne.jp

Shimazaki Naoto shimaz-n@yaa.yokogawa.co.jp

Half-Duplex Spread Spectrum Networks

Darryl Smith, B.E., VK2TDS
POBox 169
Ingleburn NSW 2565
Australia
VK2TDS@ozemail.com.au

ABSTRACT:

This paper is a response to the presentation of the TAPR SS Modem at the 1997 Digital Communications Conference in Baltimore, MD. At this conference, topology's were proposed for use of the SS radios and modems in a network, which the author of this paper feels are rather limiting. This paper proposes to extend the topology's available allowing implementation of a network rather than a collection of communicating nodes. This paper also builds on a number of ideas brought up in the authors undergraduate thesis.

Introduction

Expansion of radio based networks in amateur radio is process that is tied deeply to the technology used on the network. Packet radio links using FM radios succeeded because of the ability to incrementaly expand the network. To add another link, all that was needed was the hardware at the far end to be installed. In most cases, the link could be using existing hardware sharing time with existing links.

Put another way, amateurs find it much easier to set up one new station that two. This is especially the case when the equipment required for each station is quite expensive. This paper attempts to put the idea that a Spread Spectrum (SS) network can be designed to operate in a way that allows easy ad-hoc expansion. This paper addresses many of the problems seen in the protocols proposed for the forthcoming TAPR SS Radio.

Assumptions.

There are several basic assumptions made in this paper about the operation of the TAPR SS Radios:
- The system transmits data in 'TIMESLOTS' which are on a particular frequency for a particular period of time. During a timeslot, the frequency of the station does not change. After each timeslot, the frequency in use changes.
- That radios transmit in equal length timeslots - regardless of the amount of information to be transmitted.

- That stations throughout the network can keep track of timeslots through some absolute method (Averaged timings from adjacent stations or locked to a GPS based clock are two options)
- That it is possible for a station to hear a station that is not the closest station. That is the classic CDMA near-far problem doe not apply here. (This assumes that both stations are not transmitting on the same frequency in the same timeslot)

In the 1997 DCC two possible modes of operation were proposed for the new TAPR SS radio modem. These modes were a point to point link and a star network as shown in figure 1.

Figure 1

Point-to-Point   Star
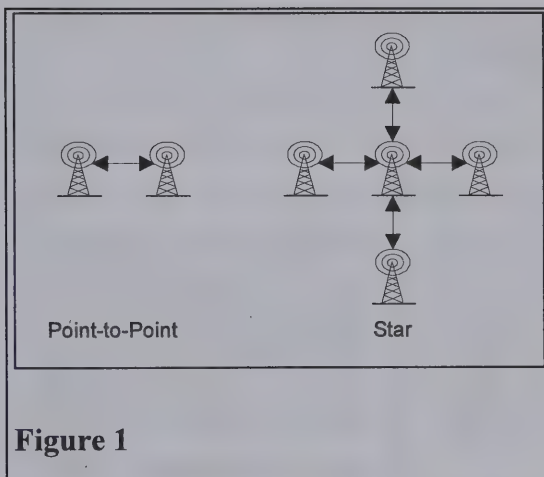
However in a spread spectrum situation this is not a good use of resources. This if especially so in the case of the star configuration. The utilisation can be defined as the time spent by all stations transmitting or receiving divided by the total time. In the start configuration with four stations, the utilisation becomes 16/40 or only 40%. This means that on average 60% of each stations time is being wasted.[1]

A network of Point-to-Point links would be ideal allowing for 100% utilisation, but this would require excessive infrastructure of a network was to be developed. All the stations in Figure 1 would share a common hopping sequence.

Compare this with a slightly smaller idealised situation on the next page . It is assumed that synchronisation can be maintained at all times. In this case the utilisation is 100%[2]. This layout is somewhere between a series of point-to-point links and a totally ad-hoc network.

---

[1] A Star configuration with 5 total stations would need a timeslot for each station to send data to the central node, and another receive data from the central node. Eight transmission timeslots are required in total. This translates to eight transmitting timeslots and 8 receiving timeslots. During the 8 timeslots the 5 stations have a total capability of 40 timeslots to transmit or receive. $(8+8)/(8*5) = 0.4 = 40\%$. A star network with 4 total stations would have a utilisation of 50%.

[2] The utilisation in this case is most easily computed by examining the amount of time that any radio is not transmitting or receiving. Since no time is wasted, the utilisation is 100%.
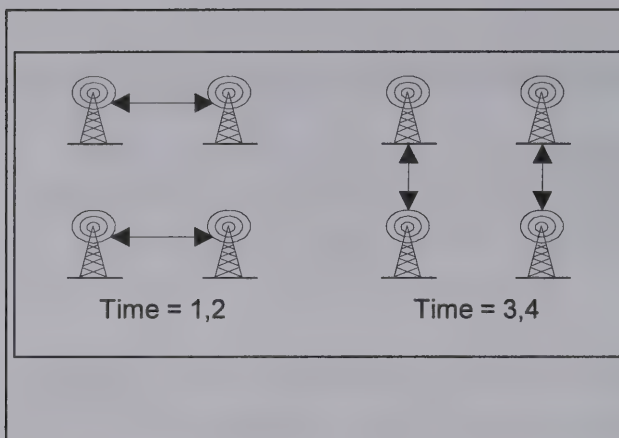
In this situation, pairs of stations would share hopping sequences. For timeslots 1 and 2, the upper pair of stations would share a hopping sequence. The lower stations would also share a different hopping sequence. In Timeslots 3 and 4, the stations on the left would share a hopping sequence, and the stations on the right would share a different hopping sequence.

Unfortunately such a topology does not scale well. In a real network we get a situation more like the one in figure 3. For a spread spectrum system to operate effectively and to be scalable, it must be able to cope with such a network.
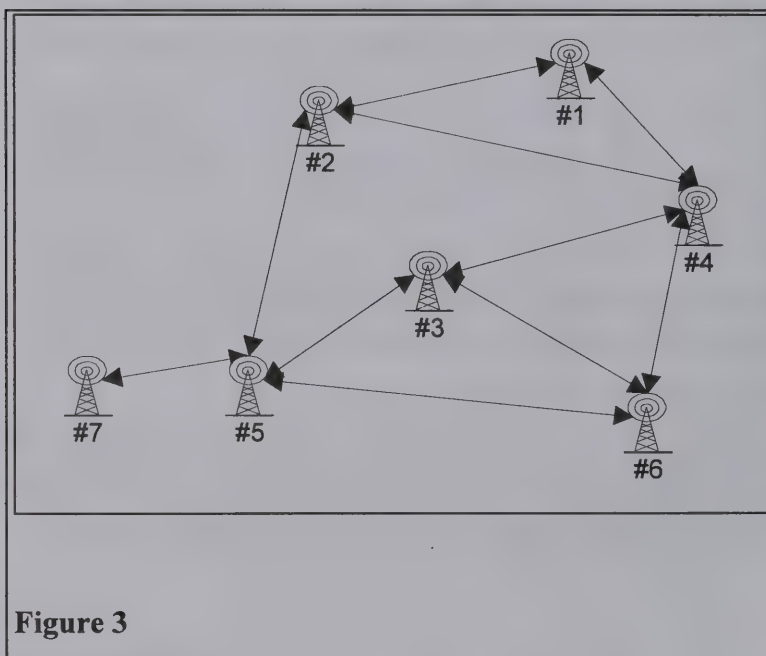
Figure 3

It can be seen that most of the time pairs of stations can communicate without any problems. However when stations 5 and 7 are communicating not all of the remaining 5 stations may communicate. This problem can only be minimised, but never eliminated.

In effect keeping the Assigned Timeslots number of unavailable stations as low as possible is one constraint for minimum energy routing.

If we are to realistically implement system as disorganised as the one in figure 3, we should look at a number of ideas.

Hopping Sequence
Each station in the network will share the same hopping sequence. Each station would be assigned a unique offset from the start of this hopping sequence, so that simultaneous transmission from all stations would be orthogonal.

Idle Mode

Each station should be listening for packets using their default hopping sequence to determine the frequency to monitor during each timeslot.

Transmitting Mode

During transmissions, the frequency used should be the frequency assigned to the receiving station for that particular timeslot.

By using just these rules to develop a network, we can see the efficiency of the network approach the 39% of Slotted Aloha.

But with any system, there is some information which is more important than others. There also tends to be a base loading and then peaks. It seems reasonable to design a network to cope with these aspects. I have therefore determined that timeslots should be coordinated between stations to reduce contention for some resources.

Routing and Time-slot Assignment

It has been shown that if power was controlled in a network, and if minimum energy routing were used, then a spread spectrum network is infinitely expandable. In the following section I have assumed that the layer above has determined the path that a packet will take. That leaves the stations just needing to work out how and when to send packets.

I propose that timeslots be assigned in a number of ways

FIXED

Periodically each station should have the opportunity to exchange information with it's neighbours, including data and planned timeslot assignments. By fixing some stations to timeslots the minimum information the network can transfer is increased.

ASSIGNED OR POLLED

During a stations fixed timeslot, it may request a number of additional timeslots over a period of time. On it's next transmission, a packet would be sent to the requesting station listing timeslots for use.

SLOTTED ALOHA

Each station will list some timeslots as being for Slotted Aloha use. These timeslots are transmitted to such as in Slotted Aloha. There is no way that other stations can determine if they are getting through, or blocking other stations dropping the maximum utilisation to 39%. However some traffic is so random that this will be the most efficient transmission mechanism.

Conclusions

In this paper I have not attempted to look at how timeslots are actually assigned and re-assigned, or how new stations are registered. I have not looked at routing protocols, but rather what happens when a decision on routing is made. I have attempted to show that some Spread Spectrum topology's are not as efficient to network scalability as others. I have also attempted to present a basis for further work on this subject.

I should point out once again that having a scalable network is essential for a spread spectrum network to operate. Without scalability, the effort is wasted. As was shown when FidoNet was introduced, a network of short links can work..

PropNET: A Proposal for an APRS-based Propagation-Research Tool

Evhen Tupis, W2EV
Chairman, Rochester (NY) VHF Group (http://vhfgroup.rochesterny.org)
17050 LaDue Road
Holley, New York 14470-9736
Evman@ix.netcom.com

PropNET Website: http://www.greeceny.com/propnet

When does the band open? Over what paths did the opening occur? Was it open to more than one location at the same time? Wouldn't it be nice to have been alerted to the opening, while it was in-progress? Wouldn't it be nice to "log the opening", and "re-play" it at a future time? With a minimal investment in equipment and software, you may be able to answer these questions for yourself!

Introduction

Automatic Position Reporting System (APRS[1]) protocol utilizes unconnected AX.25 packets from Terminal Node Controllers (TNC's) to beacon data. To date, APRS is used effectively to broadcast data such as weather information from data-ready weather stations and, when attached to a Global Positioning System (GPS), mobile stations may be tracked – in real time – on maps displayed on computer screens. Even balloons can be equipped with similar electronics, making their tracking easy.

Yet, the fact of the matter is that a vast majority of APRS stations are home or fixed in location. This is clearly evident in the screen-shot shown on this page. Only 4 moving-station icons appear on this shot of 144.39-MHz APRS activity in Western New York and Southern Ontario, Canada. The rest of the station-icons belong to home and remote-stationary weather stations. Unless one lives in Southern California, it is doubtful that anyone would ever see much movement from those stations[2]. For the most part, watching an APRS screen can be akin to watching your antennas oxidize. Until now.

---
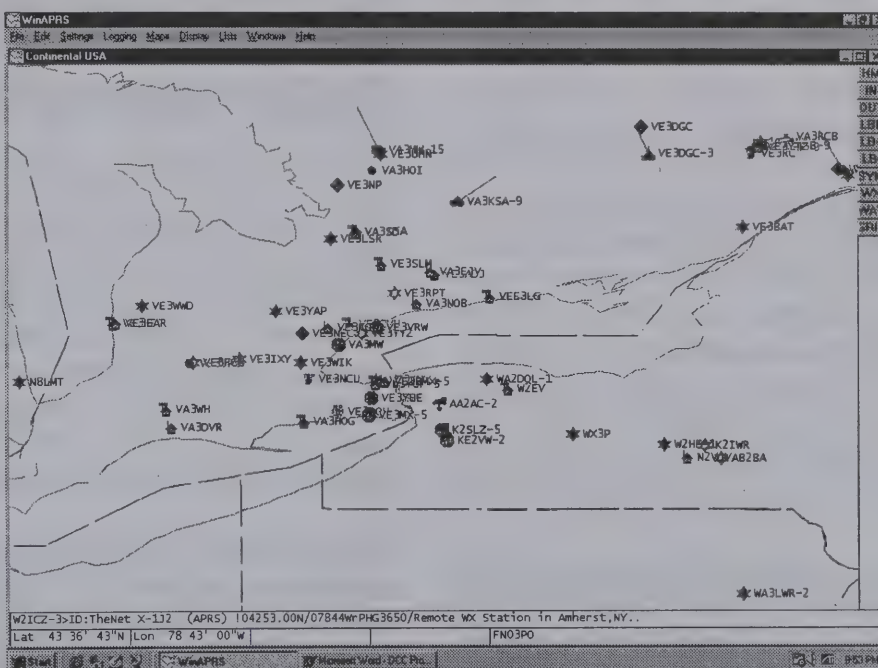
[1] APRS is a trademark registered by Bob Bruninga WB4APR, WinAPRS and MacAPRS is a trademark registered by Keith Sproul WU2Z

[2] A witticism that was plagiarized from Keith Sproul, WU2Z himself

Enter – Project: PropNET

One of the features that APRS includes is the ability to trace the path that a received-packet took, in its' journey[3]. The process is quite simple. One uses the mouse to select an icon that has appeared on your screen. You then press the "t"-key[4]. The computer then "t"races the path, for your eyes to behold. The screen-shot on the right is an example of invoking the "t"race function, tracing the packet-path between WA3LWR-2 and W2EV. Note that there is no direct path between these two stations, as the packet "hopped" through other TNC's in order to be received.

The facts imply that a vast majority of stations are home or fixed in location (yawn). So why not begin APRS operation on a band that is known for long-distance anomalous propagation, optimizing TNC configurations for channelized beacon-type operation with minimum channel-loading? Thus was borne PropNET.

I decided to immerse myself in the world of APRS, to learn the subtleties of the program, to see if there were any other features that could be pressed to PropNET use. Indeed, there were. APRS has the ability to display icons for stations only if they were received *directly* (without being digipeated). The program may also be setup to run in the background, and provide the receiving station with an audio alert if a new, DX station were to be received. Another very powerful function of the APRS system is its ability to log all of the activity on the channel, and play it back at a later time!

The push for PropNET was on. There were still several "management" or "non-technical" issues to be resolved. First, determining which band to

put it on. Another would be to determine the frequency of operation in the band. An equally close third issue is how to configure the TNC's to properly load the channel if/when the band was to open.

A Thumnail Sketch of the Network

There are three proposed "classes" of PropNET stations: Hub, Peer and Client. Hub-class stations are the equivalent of "Wide-Area Trace Digipeaters" in the APRS world. They are digipeaters placed at altitude (building top, mountaintop, etc.), with high power (100+ watts – the more, the better – really!) and good "ears". Hub's are given "channel priority" and beacon frequently (every 5-minutes or less). Peer-class stations are "almost everyone else". They are stations, which are attached to computers running a version of APRS software, patiently awaiting a band opening to report. They are given "lower channel priority" than Hub stations.

As of the time of this writing, only one TNC manufacturer makes TNC's that are capable of processing PropNET-tracable packet frames: Kantronics[5]. Therefore, in order to participate, *fully*, in PropNET, it is important that you use one of their TNC's (with ROM version 8.3 or above).

One may participate in PropNET even without a Kantronics TNC. Doing so classifies you as a

---

[3] The "Trace Command" works properly only if all TNC's in the path are configured to take advantage of this function. On the 2-meter band, less than half of the operators have configured their stations to do so as of the date of this publication.

[4] This is a command specific to the Windows version of the software, with which I am most familiar

---

[5] Kantronics and KPC are trademarks of Kantronics Corp.

Client-class station. Client's have lowest channel-priority and cannot relay PropNET frames. They can, however, beacon and receive PropNET frames yet, they don't contribute to the network's information-passing peer-to-peer infrastructure.

Visualizing PropNET

For those who have little, if any, experience with APRS software, the concept of PropNET may be a difficult one to visualize. Providing a "tutorial" of the concept, in a print publication, is an equally daunting task. Yet, before one can be expected to buy-in to PropNET, it is important that one can mentally visualize the potential power of the proposed system.

The following illustrations will attempt to outline the concept of PropNET, using manufactured screen-shots from WinAPRS[6].

Below is a representation of the Western New York PropNET "cell of activity".

To differentiate between 6-meter and 2-meter icons, I propose that Hub-class stations identify themselves with a Blue-6[7] icon. Some day, the 2 and 6-meter networks may be bridged together. It

---

[6] WinAPRS (as well as DOSAPRS and MacAPRS) may be downloaded from the Tuscon Amateur Packet Radio website: http://www.tapr.org

[7] As this goes to press, there is discussion about using a green-star icon (similar to 2-meter APRS digipeaters), but with a "6" overlayed, signifying a 6-meter Hub

will be important to differentiate icons that originated on a specific band. In this 6-meter PropNET example, Hub-class stations are Blue-6's and Peer-class stations are Diamonds.

Imagine that you are at W2EV, viewing this map on your computer, as it runs WinAPRS. You have the desire to know the path that your frames are taking to/from W2UTH. The process is quite simple. You use your mouse to point-and-click on the icon for W2UTH. Once selected, you press "t", for trace, and this is what you see:

There is no direct path between the two stations; packets are being relayed through the W2AAA-1 hub.

Just then, you hear your computer "beep" at you – not just once, but several times. That is the tell-tale indication that several new icons have appeared on your screen yet, you see no change on the map; so you "zoom out" to get a wider perspective. You are greeted with the following:

The band just opened! Three Hubs have just popped on your screen. Using your mouse, you point-and-click on KR4YL[8], press the "t"-key and the following trace appears on your computer screen:

It looks like the band just opened between central florida and Southern Ontario; the packet came from KR4YL through VE3NCU, who digipeated it to W2EV!

Hey, look at the Hub on Long Island, New York. That's too far to be groundwave, I wonder how we're hearing it. Point-and-click on WA2GUG's Hub, press the "t"race key, and you get this interesting path-map:
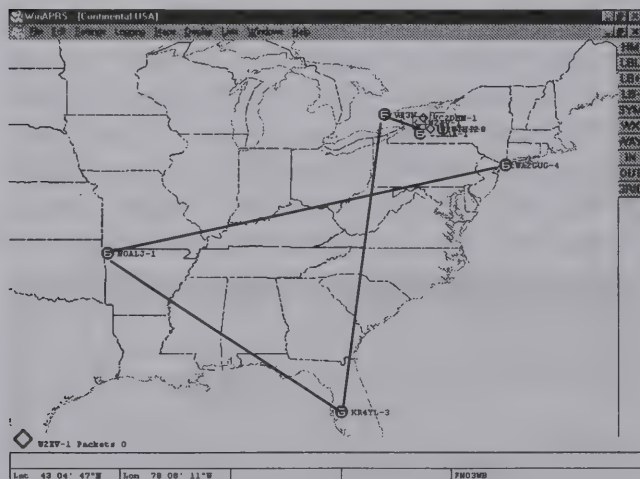
WOW! What a thought provoking graphic. Under normal circumstances, a station in Ontario, Canada would be aware only of the opening to Florida. The station in Florida would know of the opening to Ontario, Canada and to Northwest Arkansas. The station in Arkansas would know of the opening to Florida and Long Island. Yet, each would be oblivious to the anamolous propagation being experienced in the others' domain!

The potential power of PropNET is enormous! Couple this with the ability to "log" openings to disk and play them back on demand, and one's mind boggles (ok...so maybe I'm overstating a bit, but it's easy to get excited about this concept, don't you agree?).

Is PropNET for you?
Are you a pioneer? Can you work collaboratively with other enthusiasts, being flexible as to how your station is configured – and willing to make changes for the good of the network as a whole, as need dictates? PropNET may, indeed, be just what the doctor ordered.

The scope of this article was intended to outline a vision for this new experimental service. There are many technical and political issues yet to resolve. To stay up-to-date on the technical issues of PropNET, visit the website[9]. As this is going to press, I've gotten communication from Greg Jones – WD5IVD (of TAPR fame), who has agreed to host a PropNET listserve on the TAPR website, to keep us "pioneers" speaking with one voice.

Does PropNET work? No one can say one way or the other – yet. As of press time, only three stations are known to exist, but excitement is building. It is our intent to continue to build the network over the Fall and Winter, so that we can boast a "critical mass" of 6-meter PropNET stations by the time that the 1999 summer Es season kicks into high gear. Join-in as we attempt to make a little history of our own!

---

[8] KR4YL and VE3NCU, although active on APRS, are not active on PropNET – yet. :o) W2UTH is used for illustration only.

[9] http://www.greeceny.com/propnet

APRS™ and PropNET: Potential Tools for Collaborative Radio Propagation Research

Scott E. Parker, K7LU
Occidental Radio Sciences
K7LU@qsl.net

Key words: APRS, Propagation, PropNET, Six meters, Space weather.

PropNET is an APRS network operating on the six meter band whose primary objective is to monitor and study radio propagation. There is interest outside (as well as within) the amateur community in the long distance propagation modes that PropNET has been designed to study. The possibilities for collaborative research and potential benefits of inter-community cooperation are explored.

Introduction

The potential of the Automatic Packet Reporting System as a tool for monitoring and studying radio propagation has been recognized since the early days of APRS. Bruninga [1995] described APRS networks operating at HF as a "poor man's chirp sounder". He further described how the data logging features of the APRS software extend this "sounding" capability beyond a simple snapshot of current band conditions by providing an entire day's overview of HF channel dynamics. Horzepa [1997] has called attention to the ability of APRS to alert users to band openings at VHF.

Recently, Tupis [1998] has proposed PropNET, an APRS network operating at 53 MHz whose express purpose is to monitor and provide a means for investigating propagation on the six meter band. Details of how the PropNET design has been optimized for propagation study are available in this Proceedings and on the web at http://www.greeceny.com/PropNET. In addition to being a real time indicator of band conditions, PropNET will have the ability, once widely implemented, of producing a database for long term propagation study.

DXing for the non-amateur

Amateurs are not the only users of the VHF spectrum interested in using natural (ionospheric, tropospheric) modes of medium and long distance propagation. For example, military users find these modes attractive for a variety of reasons, both strategic and practical [Lott, 1997]. But in order to make good use of long haul VHF, these users must be able to accurately characterize and forecast propagation conditions. The current state of understanding and forecasting ability does not measure up to the requirements of these users in many instances. Their ability to fully exploit these long haul modes in the future will require further research in the area of VHF propagation.

Data collected by a network such as PropNET, designed explicitly to aid in advancing the state of radio propagation science, can potentially make a significant contribution toward increased understanding and improved forecasting abilities at VHF. Herein lies a unique opportunity for amateurs to make a significant contribution to the science of radio propagation and forge a symbiotic relationship with the professional research community.

Six Meters

The six meter band supports a host of interesting propagation modes. In fact, almost any mode of 6m propagation beyond line of sight tends to be categorized as "unusual". These range from F2 reflection, mundane at HF but considered unusual at VHF since it signals an MUF well beyond normal bounds, to the exotic and less understood modes such as ionospheric scatter and transequatorial ducting.

Less understood means less predictable. Several modes of long haul propagation, utilized by amateurs and potentially useful to occupants of the adjacent VHF spectrum, are known to be operative on the band, making it fertile ground for long haul propagation research. The goal of such research is an improved ability to forecast band openings, not only when the band will open but to where, for how

long and how reliably. This is a capability that amateurs would love to have and many non-amateurs absolutely require on the VHF bands. Of late, members of the scientific community have emphasized lines of research which will contribute to efforts at improving this forecasting capability.

Space Weather

The use of communications and navigation technologies whose accuracy and effectiveness are subject to conditions in the space environment is on the increase. This implies an ever increasing need to understand, characterize and predict the state of the space environment. The region of interest extends from the sun to near the earth's surface, taking in the ionosphere and its sources of variability [WG/NSWP, 1997].

In order to meet the demand for more effective mitigation of technology's vulnerability to the space environment, research activities in the field of atmospheric and space science have been redirected in recent years to focus on space weather. The goals of space weather research include augmentation of our ability to monitor, characterize and forecast the state of the space environment.

Space weather research has and will continue to provide several elements essential to advancing our ability to better understand and predict VHF ionospheric openings. Data on the state of the ionosphere and other regions of the space environment which affect it are increasing in availability and timeliness. And improvement of our ability to predict the future state of the ionosphere, a fundamental goal of space weather science, is the key to reliable propagation forecasts.

PropNET's potential contributions

How can PropNET contribute to advancing the science of VHF radio propagation forecasting once the network has been deployed on a broad scale9 Two examples come immediately to mind.

First, PropNET data can help point out potential avenues of inquiry. Space weather researchers avail themselves of data from many different sensors which measure different parameters of the space environment: electromagnetic and particle fluxes from the sun, chemical composition, dynamics and temperature structure of the upper atmosphere and electrical currents in the ionosphere and beyond. Any strong correlations between data from these sensors and band openings recorded by the PropNET database can be used as initial clues as to possible cause and effect relationships which should be investigated.

Should the pursuit of one of these initial clues prove fruitful, the end product of the research will take the form of a theory and/or propagation model. Here PropNET can contribute again. This time the network's role is theory and model verification. Are the theoretical predictions borne out in the PropNET data? How good is the model at forecasting what PropNET will see?

These are only two examples of many possible contributions that PropNET can make to collaborative research. The possibilities go beyond the basic science of radio propagation to include engineering issues such as optimization of equipment and communications parameters to better take advantage of different propagation modes.

Return on the investment

The current state of the art in tropospheric weather forecasting is much more advanced than its space weather counterpart. Not only are the relevant theories and models more accurate, but the troposphere is much more accessible to measurement. There is simply a lot more information on the troposphere, both its present state and predictions of its state in the near future, available at any given time. Amateurs have used this information to advantage in predicting tropospheric openings at VHF and above [e.g., Pocock, 1985]. Current capabilities for forecasting ionospheric openings on the VHF bands lag far behind.

The climatology of the VHF ionospheric modes has been known for some time [Jacobs and Cohen, 1982]. This is to say that we know, based on past history, when and where the different types of openings are most likely to occur. But the next generation of predictors will go beyond the seasonal or monthly probabilities that these statistical histories provide. Instead, they will forecast openings in the coming hours based on the current state of the space environment.

The amount of space weather data available to the general public, primarily through the internet, has increased dramatically in recent years. But in order to use this data in forecasting band openings, reliable physical models of the propagation modes and the ionospheric weather they rely on must be developed and refined. This is the goal of space weather research in the area of radio propagation.

112

When that goal is attained, enhanced forecasting capability for the ionospheric modes at VHF will be the dividend returned on the amateur community's investment of data into the PropNET database.

Future Directions

PropNET is currently in a formative stage. The current design of the network and of the APRS software allows network nodes to monitor propagation in real time and maintain a short term local database of propagation conditions. However, the construction of a centralized, long term propagation database is a very real possibility whose potential utility and benefits have been outlined above. Now is the time to give serious thought to the details of a central PropNET database so that the relevant issues can be resolved by the time wide scale implementation of the network has been realized. The following are some of the forefront concerns that need to be addressed:

Database Structure. The database should be designed so as to maximize its utility within both the amateur and professional scientific communities. The database should be kept as compact as possible while at the same time maintaining adequate temporal and spatial resolution.

Data Collection and Access. A plan for collecting data from individual PropNET nodes into the centralized database and for making the database accessible to potential users must be implemented.

Data Logging. The most important contributors to the database will be the PropNET hub class nodes, many of which will be unattended. Steps to facilitate automated data collection and submission by unattended sites need to be taken. This will likely involve some software design and development.

Conclusions

The possible contributions of the PropNET network proposed by Tupis [1998] to the science of VHF radio propagation have been explored. A six meter propagation database generated by the network could form the basis for collaboration between the amateur community and radio science professionals. As the PropNET network is presently in a very early stage of realization, measures necessary to facilitate the construction of a centralized propagation database will never be easier to implement. These data recording and archiving features should be integrated into the network as soon as possible.

References

Bruninga, Bob, WB4APR, "Automatic Packet Reporting System (APRS)", from the ARRL 13th Digital Communications Conference, reprinted in *Packet: Speed, More Speed and Applications, 1 Ed.*, American Radio Relay League, Newington, CT, pp. 4-1 - 4-9, 1995.

Horzepa, Stan, WA1LOU, "Chasing DX with APRS", from *QST*, 80:10, reprinted in *Packet: Speed, More Speed and Applications, 2 Ed.*, American Radio Relay League, Newington, CT, p. 1-15, 1997.

Jacobs, George, W3ASK and Theodore J. Cohen, N4XX, *The Shortwave Propagation Handbook, 2nd Ed.*, CQ Publishing, Hicksville, NY, pp. 129 - 144, 1982.

Lott, CDR Gus K., "Space Weather and U.S. Navy Requirements", *Proceedings: Space Weather Effects on Propagation of Navigation & Communication Signals*, Northwest Research Associates, Bellevue, WA, pp. 241 - 259, 1997.

Pocock, Emil, W3EP, "The Weather that Brings VHF DX", *QST*, 69:5, pp. 11 - 16, 1985.

Tupis, Evhen, W2EV, "PropNET: A Proposal for an APRS Based Propagation Research Tool", *this Proceedings*, 1998.

Working Group - National Space Weather Program, *The National Space Weather Program: The Implementation Plan*, Office of the Federal Coordinator for Meteorological Services and Supporting Research, Silver Spring, MD, 1997

Alpha-test report of PRUG96 High speed radio link

Shingo Watanabe, JG8OOM
Packet Radio User Group of JAPAN
P.O. Box 66, Tamagawa, Setagaya-ku, Tokyo 158-0094 JAPAN
nabe@mobile.icc.titech.ac.jp, nabechan@kbc.gr.jp

Satoshi Funada, 7M3LCG
Packet Radio User Group of JAPAN
P.O. Box 66, Tamagawa, Setagaya-ku, Tokyo 158-0094 JAPAN
sfunada@cs.titech.ac.jp

Kunihiko Ohnaka, 7K4JBX
Packet Radio User Group of JAPAN
P.O. Box 66, Tamagawa, Setagaya-ku, Tokyo 158-0094 JAPAN
kuni@momo.rim.or.jp

Abstract
The PRUG96 system is designed to create a reliable high-speed ham radio based computer network. This report describes a PRUG96 system using IP network protocol. We have an alpha-test structure to make it clear the weakness point of the system. The difference in the daily data throughput in various environments and error rate trend were measured.

1. Introduction

The Prug98 system is designed to create a reliable high-speed ham radio based computer network.

This report describes a PRUG96 system using IP network protocol. Use of other network protocols are expected to make the system match the radio link characteristics.

We have an alpha-test structure to identify the weak points of the system.

Expected cause of weakness.

IP network protocol is used. The network protocol isn't designed to match the radio link character and results in an unreliable link caused by radio phasing, multi-pass, hidden terminal and other problems.

High-speed radio is used. High-speed radio is always expected. High speed is a magic word, but high speed and reliability do not exist at the same time.
A newly designed Spread Spectrum 808kbps radio on 2.4GHz ham band was used.

Data measured. The difference in the daily data throughput in various environments and error rate trend were measured.

2. Structure of the Alpha Test

The test structure consists of five (you've got six stations listed) radio stations in southern Tokyo. Their antenna highs are listed for reference.

JG8OOM	30m	Rooftop of a ten-story building
JL1ZCF	6m	Rooftop of a two-story house
7K4KES	5m	
7K4JBX	9m	Rooftop of a three-story house
JM1ZNW	8m	Rooftop of a two-story house

This test environment is similar to practical use. Not all the stations are line of sight. Hidden terminals do exist. The distance between stations are from 200m to 2000m. Error rate is not always of a level necessary to maintain a lasting link.
Our PRUG96 system provides automatic routing system. The routing table changes dynamically.

3. Throughput

FTP Throughput. Throughput numbers described here are measured by the FTP application. Note that the real throughput of radio porting is much better. The radio handles data and large overhead for better error correction.

Best throughput

Tables 1 through 4 show the best results. They show approximately 10kbps per second throughput. Which is better than ISDN (64kbps) or High-speed modem (56kbps) links. It must be noted that the same test in a room resulted in up to 15kbyte per second.

File Size	File Transfer Time	File Transfer speed
920504 byte	84.45 sec	10.64 Kbyte/sec
920504 byte	122.96 sec	7.31 Kbyte/sec
920504 byte	89.41 sec	10.05 Kbyte/sec
384124 byte	28.63 sec	13.10 Kbyte/sec
384124 byte	?	12.86 Kbyte/sec

Table 1: 7K4JBX -> JG8OOM

File Size	File Transfer Time	File Transfer speed
920504 byte	157.12 sec	5.72 Kbyte/sec
384124 byte	50.97 sec	7.36 Kbyte/sec

384124 byte	38.49 sec	9.75 Kbyte/sec
384124 byte	44.70 sec	7.55 Kbyte/sec

Table 2: 7K4JBX <- JG8OOM

File Size	File Transfer Time	File Transfer speed
128640 byte	10.83 sec	11.60 Kbyte/sec
384124 byte	36.74 sec	10.21 Kbyte/sec
384124 byte	52.52 sec	7.14 Kbyte/sec
384124 byte	35.65 sec	10.52 Kbyte/sec

Table 3: 7K4JBX -> 7K4KES

File Size	File Transfer Time	File Transfer speed
384124 byte	56.68 sec	6.62 Kbyte/sec
384124 byte	35.26 sec	10.64 Kbyte/sec
384124 byte	29.68 sec	12.64 Kbyte/sec

Table 4: 7K4JBX <- 7K4KES

Under Interference

Tables 5 through 7 show the influence of concurrent access to one FTP server. The Difference between two equal application users is expected to be caused by radio interference between these two users. JG8OOM is located on the top of high building vice 7K4KES, who is in a residential area. This result may be explained by 7K4KES's receiver blocking caused by JG8OOM's transmission. JG8OOM and 7K4KES are often thought of as hidden terminal in relation to each other.

File Size	File Transfer Time	File Transfer speed
384124 byte	a) 29.75 sec b) 68.07 sec	12.61 Kbyte/sec 5.51 Kbyte/sec
384124 byte	a) 33.32 sec b) 33.32 sec	11.26 Kbyte/sec 5.79 Kbyte/sec
384124 byte	a) 26.61 sec b) 54.62 sec	14.10 Kbyte/sec 6.87 Kbyte/sec

Table 5: a) 7K4JBX -> JG8OOM, b) 7K4JBX -> 7K4KES

File Size	File Transfer Time	File Transfer speed
384124 byte	a) 45.25 sec b) 88.49 sec	7.60 Kbyte/sec 4.29 Kbyte/sec
384124 byte	a) 29.44 sec b) 66.88 sec	12.74 Kbyte/sec 5.61 Kbyte/sec
384124 byte	a) 27.43 sec b) 64.90 sec	13.68 Kbyte/sec 5.78 Kbyte/sec

Table 6: a) 7K4JBX -> JG8OOM, b) 7K4JBX <- 7K4KES

File Size	File Transfer Time	File Transfer speed
384124 byte	a) 73.23 sec b) 81.49 sec	5.12 Kbyte/sec 4.60 Kbyte/sec
384124 byte	a) 85.49 sec b) 72.74 sec	4.39 Kbyte/sec 5.16 Kbyte/sec
384124 byte	a) 68.84 sec b) 46.06 sec	5.45 Kbyte/sec 8.14 Kbyte/sec

Table 7: a) 7K4JBX <- JG8OOM, b) 7K4JBX -> 7K4KES

Tables 8 through 9 show two pairs of server/client FTP transfer throughput. It means that there was almost equal resources distributed to each pair.

File Size	File Transfer speed
384124 byte	a) 7.79 Kbyte/sec b) 4.60 Kbyte/sec
384124 byte	a) 3.67 Kbyte/sec b) 5.45 Kbyte/sec
384124 byte	a) 3.49 Kbyte/sec b) 6.26 Kbyte/sec

Table 8: a) 7K4KES -> 7K4JBX, b) JG8OOM -> JL1ZCF

File Size	File Transfer speed
384124 byte	a) 5.04 Kbyte/sec b) 4.44 Kbyte/sec
384124 byte	a) 8.50 Kbyte/sec b) 4.75 Kbyte/sec
384124 byte	a) 5.25 Kbyte/sec b) 4.51 Kbyte/sec

Table 9: a) 7K4JBX -> 7K4KES, b) JG8OOM -> JL1ZCF

4. Fluctuation of error rate

Figure 1 shows error rate fluctuation on a specific day. Error rates change periodically. The rate falling in morning exists daily. Another dip often arises early in the evening, especially between 16:00 and 18:00. The cause is not clear, but it is possibly as stated below:

a. Microwave oven. Many ovens are active during mealtimes. This particular phenomenon may by explained by effect caused by the microwave oven's radiation.

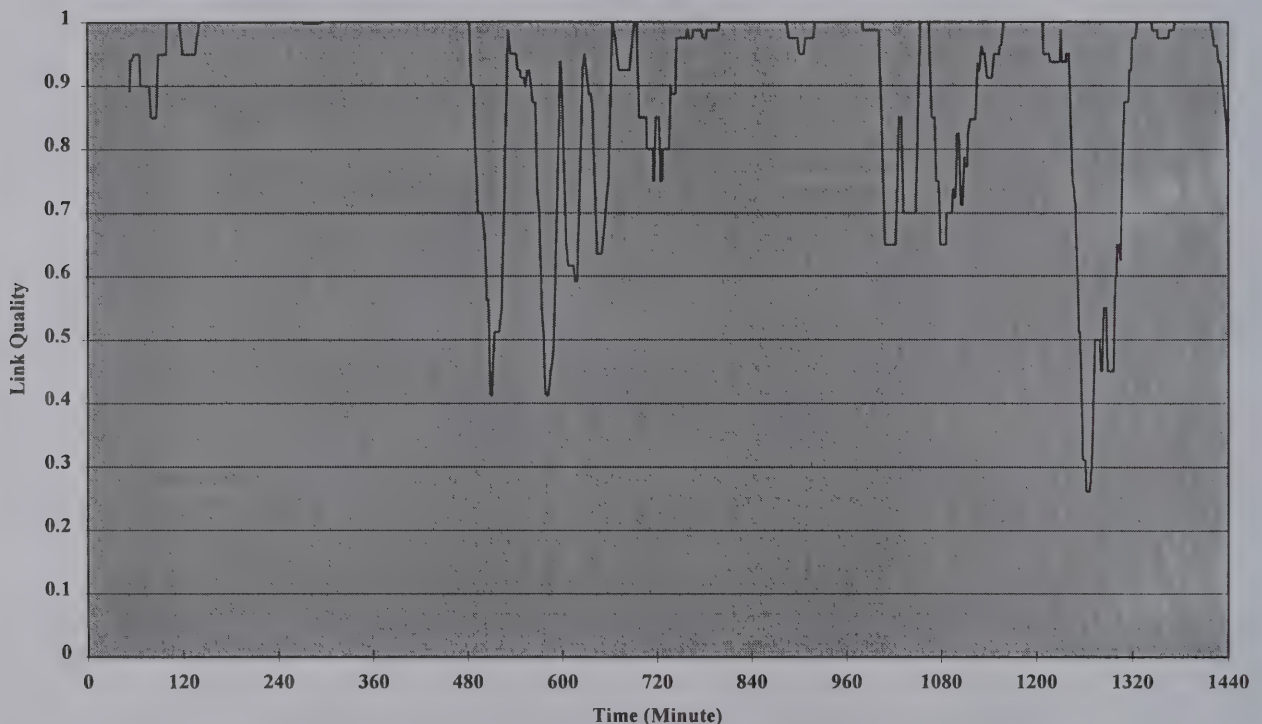b. Traffic. Good rates were achieved at midnight. Vehicle activity may responsible to the radio link failure.

Figure 1: Fluctuation of error rate

5. Conclusion

Practicality assessment

(1) Application throughput was found to be competitive with commercial high-speed data communication services on wire.

The field test showed that up to 14kbyte per second throughput on FTP transfer rates were achieved.

(2) Automatic routing function

Automatic routing is a principal characteristic of the PRUG96 system. The distributed routing function will make this system practical on dynamic changing data pass in ham radio networks.

(3) Radio link fluctuates even on relatively short distance.

This weakness is expected to be solved using an additional power amplifier. A 20-25 dBm output amplifier has been designed and is currently under testing.

(4) Hidden terminal problem.

The alpha test was based on mixed protocol layers with the newly designed routing function and existing functions (i.e. CSMA).

Mismatching between existing protocol layers and radio physical layers will be solved with the implementation of a newly designed media access layer, network layer, and transport layer. PRUG is now designing plural original media access layers and is researching higher layers.

In the future. Error rate trends over long periods will be measured.
We are interested in studying the relationship between data error rate and weather conditions and the day of the week.

On-air Measurements of CLOVER II and CLOVER 2000 Throughput

Ken Wickwire (KB1JY), Mike Bernock (KB1PZ) and Bob Levreault (W1IMM)

1. Introduction

This paper is one of a series treating on-air measurement of throughput in eight-bit characters per second (cps) for various HF data-transmission protocols of interest to amateurs (see the references to our other reports at the end of the paper). Here we describe an extensive set of measurements of throughput for compressed and uncompressed text files sent over near-vertical-incidence-skywave (NVIS) and one-hop skywave (OHS) paths. The on-air tests used the CLOVER II and CLOVER 2000 waveforms and the file transfer protocols implemented in the HAL CLOVER terminal packages.

NVIS paths, which often experience relatively difficult channel conditions characterized by multipath interference, high noise and QRM, are used to communicate over 20- to 300-mile ground distances using antennas that can launch energy at high takeoff angles. OHS paths need antennas that launch at low takeoff angles. These paths are many hundreds or thousands of miles long and are often easier to communicate over than NVIS paths.

Several versions of the CLOVER modem are available (the PCI4000, DSP4100, P38, and others). They are distinguished mainly by the size of their modulation constellations (number of different possible data symbols) and by the resulting maximum speed of data transmission. The number of symbols used by a modulation scheme affects not only throughput but also dictates the processing power required of the associated hardware (computer or modem or both).

CLOVER II and CLOVER 2000, which run on the HAL PCI4000 and DSP4100 hardware platforms, were designed to process received signals in the BPSM ("binary phase-shift modulation"), QPSM (4-ary phase), 8PSM (8-ary phase), 8P2A (8-phase, 2-amplitude) and 16P4A (16-phase, 4-amplitude) modulation modes. (Because of the fact that CLOVER uses a non-standard multi-tone signaling scheme, HAL has chosen not to call the phase-shift modulations "PSK.") In ARQ (connected) operation, CLOVER chooses among these modulation modes in response to changing channel conditions.

CLOVER II, developed primarily for amateur radio applications, modulates four tones separated by 125 Hz and occupying a 500-Hz bandwidth. The faster CLOVER 2000 waveform modulates eight tones separated by 250 Hz and occupying 2000 Hertz. CLOVER 2000 was developed mainly for commercial applications but with the movement in HF from voice to data communications it may eventually be authorized for amateur use.

All of the CLOVER modems come with software (including a DOS-based GUI) that performs uncompressed and compressed file transfers using the CLOVER automatic repeat request (ARQ) protocol. Using the compressed mode, one can send graphics and other "binary" (eight-bit-character) file data, in addition to text files. The uncompressed mode is generally used for chat sessions and text-file transfers.

The CLOVER II data-transmission protocol changes its modulation mode (varying the number of bits per symbol while keeping the symbol *rate* fixed at 31.25 symbols per second). (This means that the shape of the CLOVER II signal spectrum stays the same as the modulation adapts itself to the channel, a property of all the CLOVER versions.) The four "tone pulses" of the waveform are transmitted in numerical order, with 32 milliseconds between occurrences of the same tone pulse. The CLOVER II CCIR emission designation is 500H J2 DEN or 500H J2 BEN and its maximum uncompressed throughput in the ARQ mode is 69.6 characters (8-bit bytes) per second.

CLOVER 2000 runs on special versions of the PCI4000 and DSP4100 modem platforms (it cannot run on the P38 card). Its CCIR emission designation is 2K0H J2 DEN or 2K0H J2 BEN. Although similar to CLOVER II, CLOVER 2000 operates with eight tone pulses at twice the fixed CLOVER II rate, or 62.5 symbols per second, and the order of its transmitted frequencies (tone-pulses), separated by 250 Hz, is scrambled to reduce adjacent-tone interference. CLOVER 2000 changes its modulation mode (and thus the number of bits per symbol) in a way similar to that of CLOVER II: if BPSM is chosen as the modulation (typical of a poor channel, and always used for control frames), only one bit per symbol (tone pulse) is sent and received (for a total data rate without overhead of 8 x 62.5 = 500 bits per second). If 16P4A is chosen (only on extremely good channels, with high SNRs and almost no multipath), the effective data rate without error-coding and other overhead is 3000 bits per second. The maximum uncompressed throughput in the ARQ mode for CLOVER 2000 is 249.3 characters (8-bit bytes) per second.

The CLOVER system assesses channel conditions (at receivers) by measuring the amount of error-correcting capacity [ECC] used to remove data errors, the number of erroneous data blocks and the phase dispersion [PHS, related to multipath], among other things. Receivers send this information to transmitters at turnarounds, and transmitters change their modes according to the most current channel information sent to them by receivers. This adaptability to the constantly changing channels, combined with the underlying Reed-Solomon FEC coding, and, when necessary, retransmission of erroneous frames, is the key to CLOVER's reputation as one of the two fastest and most reliable HF data communication protocols available to hams. (The other is the PacTOR II system, which we plan to test later.)

For further details on how CLOVER works, see the HAL documentation supplied with the various versions and the descriptions published by Ray Petit and Bill Henry in the *RTTY Journal*, *QST*, *QEX* and elsewhere between 1990 and 1994.

The NVIS paths we have studied, which are 25 to 140 miles long, frequently display strong multipath, high local and propagated noise, D-layer absorption at mid-day and occasionally strong interference from other stations operating in both voice and digital modes. (Horizontal antenna polarization at all our NVIS stations allowed us to be fairly certain we were using NVIS rather than surfacewave propagation, and this was confirmed by the fading we observed during many file transfers.) We measured one-hop skywave throughput on 1000-mile paths from Massachusetts and New Hampshire to Illinois. These paths produced less multipath interference than our five NVIS links and therefore in some cases somewhat higher throughput.

We tested CLOVER II in the ham bands and on assigned frequencies outside the ham bands. CLOVER 2000 was tested only on assigned frequencies. The majority of NVIS measurements were at 3.2 and 3.6 MHz, with a few at somewhat higher (up to 5 MHz) and lower (down to 2 MHz) frequencies. The OHS measurements were made at 10.3, 10.5, 12.2 and 15.7 MHz. (Amateur frequencies usually had too much QRM for CLOVER II OHS tests, and the legality of using CLOVER 2000 in the amateur bands is unclear to us.) For the OHS tests we normally avoided mid-day testing on frequencies far below the MUF. (On such frequencies there are two "windows of good performance" in the morning and evening when the MUF is either rising or falling through the operating frequency. We did most of our testing in such windows.) Output power (about 100 watts) and antennas at all stations were typical of those used by hams.

Daytime was defined to occur between the fixed times of 1000 and 2200 GMT (about 5:00 AM to 5:00 PM local time). Note that because our measurements were made over the course of nearly nine months, "nighttime" (roughly 5:00 PM to 5:00 AM local time) was not always associated with darkness at the path midpoint. Nevertheless, most nighttime measurements were taken in conditions that characterize conventional nighttime HF propagation: high noise and increased interference.

Since we achieved the goal of working near the MUF only approximately with our available frequency set (especially for daytime CLOVER II tests), our results are conservative. A properly set up adaptive automatic link establishment (ALE) system that used quasi-real-time channel assessments to choose the best operating channel for the CLOVER waveform would have allowed us to avoid a lot of the interference and "far-from-the-MUF" conditions that we occasionally tested in over OHS paths.

The NVIS tests covered the six-month period from September 1997 to February 1998. The OHS tests covered parts of the eight-month period from January 1998 to July 1998, with most of the OHS CLOVER II tests toward the end of that period. The average sunspot number during these periods was between about 20 and 70, so MUFs were in the lower half of their eleven-year up and down cycle.

The rest of the paper describes the paths between stations and layout of antennas, the HAL CLOVER interface and our file-transfer operations, the files sent, the recorded data format, statistical analysis software, a statistical summary of the data, a discussion of the statistical results and concluding remarks.

2. Layout of Paths and Discussion of Antennas

The stations used for the NVIS tests are in Bedford, Mass. (KB1JY), Norfolk, Mass. (W1IMM), Derry, N.H. (KB1PZ) and Portland, Maine (KB1JY-1). Bedford used an 80m dipole up 30 feet for NVIS tests, and a terminated, bottom-fed 125-ft longwire pointing southwest for OHS tests. Norfolk used an 80m dipole up 40 feet for NVIS tests. Derry used an off-center-fed 80m dipole up 30 feet for NVIS and OHS tests. Portland used an end-fed 100-foot unterminated longwire pointed west for NVIS links. The stations in the OHS tests are in Bedford, Mass. (KB1JY), Derry, N.H. (KB1PZ), and Urbana, Illinois (W9KVF). In the OHS tests Bedford used the resistively terminated sloping longwire running southwest, Derry the off-center-fed 80m dipole and Urbana a KLM-7 seven-element log-periodic up 75 feet. The links (followed by lengths and rough estimates of the percentage of data collected over each link) are:

<div align="center">

NVIS
Bedford-Derry (25 miles, 25%)
Bedford-Norfolk (35 miles, 25%)
Derry-Norfolk (60 miles, 25%)
Portland-Derry (60 miles, 10%)
Portland-Norfolk (140 miles, 15%)

OHS
Bedford-Urbana (1000 miles, 70%)
Derry-Urbana (1000 miles, 30%)

</div>

The NVIS links run more or less north-south and the OHS links east-west. At least 100 transfers were conducted in each throughput category.

3. The HAL CLOVER User Interface and CLOVER File Transfer Protocol

Although there are graphical user interfaces (GUIs) for running CLOVER that may be more sophisticated than the DOS-based one provided by HAL (Express, XPWIN and others), the HAL GUI is presently the best one to use for making throughput measurements because it gives the transfer time of compressed file transfers. During the course of our tests we moved through several versions of the CLOVER firmware. Some of the firmware upgrades made improvements in CLOVER's ARQ scheme. The improvements appear to have been in the nature of fine-tuning since we did not notice large increases in performance.

To send a file with the HAL GUI, one first establishes a CLOVER ARQ link with the receiving station by selecting (or entering) the callsign of the station and sending a carriage return.

Once the link is established (which is usually confirmed in a few seconds by a LINKED message on the GUI and the appearance of "HIS" channel statistics on the display), one can enter the name of a file to be transferred. The HAL software allows one to view and record channel statistics (signal-to-noise ratios, phase dispersion, etc.) during file transfers and chat sessions. These statistics allow one to make changes (between links) in the operating parameters (via the "BIAS" setting) in response to assessed channel state. They can also provide fascinating insight into the workings of a file transfer (see the references at the end for our paper on CLOVER channel statistics).

After the filename has been entered, one has a choice of sending a text file from the TX Buffer in uncompressed mode or sending any file ("Send from Disk") in compressed mode. Only generic text can be sent in uncompressed mode; files with control characters in them will abort a transfer attempt. Files sent in compressed mode can have any format, although some files may not benefit from compression, and may even be expanded slightly by the compression process. Conventional text files always benefit from compression.

Some people view the throughput of uncompressed files as the inherent, or "true" performance of a waveform, error-control scheme and file-transfer system. Others consider compression to be part of a protocol if it is provided as a constituent of the "common" interface provided with a modem's hardware. To satisfy both camps, we have measured throughput in these tests of both compressed and uncompressed text files.

To measure the throughput of uncompressed (text) files, we had to record transfer time in our transfer-data files by hand, since the HAL software does not display transfer times of data sent from the text buffer. This was not as onerous as it sounds since once we decided on a recording format we were able copy, paste and edit with a text editor to speed up data recording.

All of the files sent for this report consisted of readable text. To measure the throughput of compressed text files, we took advantage of the HAL software's calculation and display of file transfer time on the GUI screen. Through experimentation we deduced that the optimal file size (highest throughput with smallest test time) for throughput assessment of CLOVER was between 10 and 40k bytes, and most of our files had sizes in that range.

Files already compressed by other applications (for example, GZIP-ed files) may get sent via the HAL CLOVER GUI in even smaller form than those compressed only by the PKWARE algorithm used by HAL, but we did not do this in our tests. Precompression in HF data communication is nevertheless a useful technique that deserves further study.

4. Recorded Throughput-Data Format

The data archive file into which the results of each transfer were entered contains the date-time group at the time (GMT) of the transfer, callsign of receiving station, callsign of sender, the CLOVER interface (for these tests, that of the HAL DSP4100), uncompressed file size in bytes, compressed file size in bytes, transfer time in seconds, predominant waveform used by CLOVER during the transfer (AUTO = set adaptively by the HAL software), HF frequency in megahertz, observed channel condition (Q = quiet, N = noisy, etc.), the power adjustment mode (usually F = fixed), the BIAS (F = FAST, N = NORMAL, R = ROBUST) and the hand-calculated throughput in characters per second (cps). The latter doesn't have to be accurate, or even recorded, since it's calculated later by the analysis software. In the case of uncompressed files sent from the TX buffer, the transfer time is read from the computer clock and may therefore be off by a second or two. This makes little difference to throughput calculations for files that take several minutes to send.

The BIAS is a parameter that describes the error-correcting overhead (and thus the number of data-bytes per ARQ block) chosen by the transmitting station's operator for a communications session. Robust bias is usually chosen when the channel appears bad (noisy, low signal-to-noise ratios, interference) and fast bias when it appears to be good.

Here's an excerpt from the NVIS transfer-data file for CLOVER II tests run from KB1PZ to W1IMM and KB1JY in October 1997:

```
04.10.97 01:04:00 W1IMM KB1PZ 4100 20000 10176 363   AUTO   3.6155   N   F   F   55
04.10.97 01:12:00 W1IMM KB1PZ 4100 20000 10176 375   AUTO   3.6155   N   F   F   53
04.10.97 01:19:00 W1IMM KB1PZ 4100 20000 10176 548   AUTO   3.6155   N   F   R   36
04.10.97 01:30:00 KB1JY KB1PZ 4100 20000 10176 433   AUTO   3.6155   N   F   N   46
04.10.97 01:37:00 KB1JY KB1PZ 4100 20000 10176 480   AUTO   3.6155   N   F   N   42
04.10.97 01:46:00 KB1JY KB1PZ 4100 20000 10176 384   AUTO   3.6155   N   F   F   52
```

Files like this are opened and analyzed by a data-analysis program described in the next section.

5. The Data-analysis Software

The results in the data archive were analyzed off-line by a program called summary_clo.c. This program reads the archive file line-by-line looking for various strings. As it moves through the file to the end-of-file indicator, the program keeps running totals of throughput and other data corresponding to the strings, from which it calculates statistics such as the average and standard deviation of the throughput. The statistics are written to a summary file after the pass through the archive file. Switches in the summary code are set before each run to pick out specific data (corresponding to various string combinations) for analysis. For example, we select lines with differing actual and compressed file sizes to pick out compressed file transfers, and use the date-time group to distinguish daytime from nighttime transfers. Since the summary program was written to analyze archive files of fixed format but arbitrary length, summaries of the data collected so far can be made at any time.

Shown below is the output of the summary program for all CLOVER-II one-hop skywave (OHS) tests run from March 1998 to mid-July, 1998 (a subset of all such data). For this output we set the software switches to compute throughput statistics for *nighttime compressed text files*.

```
Statistical Summary of CLOVER Throughput Tests:
16.07.98 17:20:01 CLOVER-II NIGHTTIME COMPRESSED

NUMBER OF CLOVER XFERS IN SAMPLE = 80, CLOVER BW = 500 Hz
E(FILE_SIZE) = 22500.0 bytes, E(COMPRESSED_SIZE) = 11490.9 bytes
E(TRANSFER TIME) =   674.2 s, sd(TRANSFER TIME) =   393.0 s
E(THRUPUT) = 36.60 cps, sd(THRUPUT) = 12.44 cps, sd(mean_THRUPUT) = 1.391 cps
MAXIMUM THRUPUT = 63.90 cps, E(THRUPUT/Hz) = 0.073 cps/Hz
E(COMPRESSION RATIO) = 50.99%, sd(COMPRESSION RATIO) = 0.20%
Lowest compression ratio = 50.83%; Compressed_size:File_size = 5083:10000
Highest compression ratio = 51.39%; Compressed_size:File_size = 20557:40000
NUMBER OF UNCOMPRESSED TRANSFERS IN SAMPLE = 0
7 transfer failures; P(transfer success) = 80/87 = 0.92
```

The output shows that the average throughput for 80 uncompressed-text-file transfers was about 37 characters per second (cps) and that the largest observed throughput in this mode was about 64 cps. The sd(THRUPUT) reflects the spread of the throughput measurements about their average. Roughly speaking, about two-thirds of a set of measurements will be within one standard deviation (here 12.4 cps) of their mean and over 90% will be within two standard deviations of their mean.

We also calculate the "standard deviation of the mean throughput" [sd(mean_THRUPUT)] in characters per second and the average throughput per Hertz of signaling bandwidth. The standard deviation of the mean throughput (equal to the standard deviation of the throughput divided by the square root of the sample size) is an assessment of the variability of the mean itself (which has its own statistical variability). The sd(mean) above suggests that our sample size in this case is big enough to make us confident that if we collected many more throughput measurements under roughly the same conditions, we would not get an average throughput that differed from the one above by more than about 1.4 characters per second.

To estimate the average throughput per Hertz [E(THRUPUT/Hz)], we divide the average throughput by the average signaling bandwidth. For CLOVER II, the signaling bandwidth is 500 Hz and for CLOVER 2000 it's 2000 Hz (see the Clover documentation).

The compression ratio is defined as the ratio of compressed size in bytes to uncompressed size in bytes. A ratio of 100% therefore means no compression.

For the tests analyzed here we also kept track of the number of failed transfers and calculated the percentage of successful file transfers. Unsuccessful transfers occurred when, after a successful link, the number of times the modem tried to send a data frame exceeded a GUI-programmable limit of 20, causing the modem to terminate the link. We did not include failures to link in our transfer success ratios. In the excerpt given above, 87 transfer attempts resulted in ARQ links, seven of which were terminated by the modem when the link timed out before the file got through. This led to a transfer success ratio of 80/87 or approximaely 92%.

6 . Statistical Summary of Throughput Results

The results of our NVIS and OHS tests of CLOVER II and CLOVER 2000 (as of August 1998) are summarized in Tables 1 through 4 below. They correspond to CLOVER II NVIS and OHS transfers and CLOVER 2000 NVIS and OHS transfers in that order. The first column in each table gives the average throughput and its standard deviation, the average throughput per Hertz, the standard deviation of the mean throughput and the maximum observed throughput. The second column gives the number of transfers and the probability in percent of successful transfer [P(good xfer)] in each case. The third column gives the mean and standard deviation of the compression ratio for compressed transfers (100% means no compression). The fourth column gives the mean and standard deviation of the transfer time in seconds and the fifth column the average number of bytes in the original, uncompressed files.

Table 1. Statistical Summary of CLOVER II NVIS Throughput Data

File Type & Time	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No. Xfers P(good xfer)	E(Compr. Ratio) (CR) sd(CR)	E(xfer_tm) sd(xfer_tm)	E(No_char)
Uncompr. Text Day	34.3 cps 11.8 cps 0.07 cps/Hz 0.77 cps 69.0 cps	239 98%	—	980 s 522 s	31346
Compr. Text Day	52.1 cps 19.9 cps 0.10 cps/Hz 1.52 cps 127.5 cps	172 98%	51.4% 1.8%	704 s 360 s	33986
Uncompr. Text Night	23.8 cps 8.2 cps 0.05 cps/Hz 0.74 cps 45.7 cps	125 93%	—	1080 s 558 s	24259
Compr. Text Night	37.2 cps 12.3 cps 0.07 cps/Hz 1.15 cps 61.9 cps	115 98%	51.0% 0.3%	781 s 564 s	26977

Table 2. Statistical Summary of CLOVER II OHS Throughput Data

File Type & Time	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No. Xfers P(good xfer)	E(Compr. Ratio) (CR) sd(CR)	E(xfer_tm) sd(xfer_tm)	E(No_char)
Uncompr. Text Day	24.3 cps 7.7 cps 0.05 cps/Hz 0.67 cps 54.6 cps	131 90%	—	1049 s 461 s	24626
Compr. Text Day	42.3 cps 14.6 cps 0.08 cps/Hz 1.26 cps 87.2 cps	134 94%	51.1% 0.2%	668 s 322 s	25373
Uncompr. Text Night	21.3 cps 7.2 cps 0.04 cps/Hz 0.68 cps 36.5 cps	113 97%	—	864 s 517 s	16637
Compr. Text Night	38.3 cps 12.1 cps 0.08 cps/Hz 1.08 cps 63.9 cps	125 94%	51.0% 0.2%	615 s 347 s	21760

Table 3. Statistical Summary of CLOVER 2000 NVIS Throughput Data

File Type & Time	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No. Xfers P(good xfer)	E(Compr. Ratio) (CR) sd(CR)	E(xfer_tm) sd(xfer_tm)	E(No_char)
Uncompr. Text Day	91.2 cps 32.6 cps 0.05 cps/Hz 1.82 cps 186.7 cps	321 95%	—	527 s 345 s	45964
Compr. Text Day	166.2 cps 67.8 cps 0.08 cps/Hz 5.11 cps 333.3 cps	176 94%	51.4% 0.3%	308 s 189 s	43454
Uncompr. Text Night	58.6 cps 25.9 cps 0.03 cps/Hz 1.89 cps 148.2 cps	188 88%	—	625 s 295 s	32584
Compr. Text Night	93.3 cps 39.0 cps 0.05 cps/Hz 2.77 cps 242.4 cps	199 95%	51.3% 0.3%	474 s 353 s	39156

Table 4. Statistical Summary of CLOVER 2000 OHS Throughput Data

File Type & Time	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No. Xfers P(good xfer)	E(Compr. Ratio) (CR) sd(CR)	E(xfer_tm) sd(xfer_tm)	E(No_char)
Uncompr. Text Day	117.2 cps 37.3 cps 0.06 cps/Hz 2.04 cps 208.3 cps	334 92%	—	377 s 388 s	36008
Compr. Text Day	180.9 cps 53.5 cps 0.09 cps/Hz 4.34 cps 288.2 cps	152 92%	51.4% 0.3%	319 s 203 s	53978
Uncompr. Text Night	82.8 cps 31.0 cps 0.04 cps/Hz 2.92 cps 154.4 cps	113 83%	—	478 s 287 s	34478
Compr. Text Night	163.6 cps 54.4 cps 0.08 cps/Hz 4.81 cps 248.8 cps	128 96%	51.4% 0.2%	299 s 130 s	44167

7. Discussion of Results

CLOVER II

Tables 1 and 2 show that the inherent (no compression used) over-the-air average throughput of CLOVER II on our links is around 34 characters per second (cps) on NVIS paths and 24 cps on OHS paths for daytime operations. Nighttime NVIS and OHS throughput averages for uncompressed files are 24 and 21 cps. (The standard deviations of these mean throughputs are all about 0.7 cps, giving us high confidence that additional measurements made under the same conditions would yield nearly the same mean throughputs.)

Average CLOVER II throughputs for compressed text files are 52 and 42 cps on our NVIS and OHS paths during the day, and 37 and 38 cps at night, with standard deviations of mean throughput around 1 cps. These average throughputs, and the average compression ratios of about 51% given in column 4 of the tables, show that the HAL GUI software's PKWARE compression is squeezing our text files down to about half their original size. This, of course, roughly doubles text-file throughput. That compressed throughputs are in fact slightly less than twice the corresponding uncompressed ones is due probably to a combination of slight differences in the times files were sent (and hence channel conditions) and the additional time needed to compress and decompress the files.

At first glance, the fact that average OHS throughputs for CLOVER II are in some cases smaller than average NVIS throughputs is surprising, since our and many others' experience with these and other protocols has shown that NVIS conditions for data transmission are generally worse than OHS conditions. On NVIS paths there is *usually* more multipath, noise and interference on the band of frequencies that lie below the MUF than on OHS paths, even when comparisons are made in different seasons. However, seasonal and frequency-dependent effects of noise and interference on HF channels often overturn common experience.

In our tests, CLOVER II NVIS data were collected in fall and winter on mostly amateur frequencies that were unusually quiet during the day. However, scheduling forced us to gather our OHS data (on amateur and non-amateur frequencies) in spring and summer, when lightning noise was a feature of a long period of rainy weather in the eastern half of the country. Several of the OHS frequencies we had available in the spring were also affected by powerful broadcast stations located only a few kilohertz from our carriers. This unlucky but all-too-realistic combination of bad conditions was the reason why our CLOVER II OHS throughputs were lower than their NVIS counterparts. Experience suggests that if we had been able to collect our CLOVER II OHS data on frequencies with amounts of burst noise and interference similar to those on our NVIS frequencies, CLOVER II OHS throughputs would have been twenty to thirty percent higher than NVIS throughputs.

Given enough frequency choices, an automatic link establishment (ALE) system, such as prescribed in MIL-STD-188-141A, and now widely available, can almost always find useful, interference-free frequencies that lie below the MUF. The next generation of ALE systems, being developed now, may also do a better job than the current generation of predicting the performance of non-FSK waveforms like CLOVER.

In all cases the probability of successful file transfer with CLOVER II was 90% or higher (with the default maximum number of retries of 20). This suggests that if CLOVER II can establish a link it can almost always (even at night) complete a file transfer. (The transfer success rate given a link can be raised to almost 100% by raising the retry limit, at the expense of wear and tear on radios and amplifiers.) Although we were usually successful in choosing frequencies that supported linking, we sometimes had to give up when the MUF dropped below our set of available operating frequencies and linking became impossible. (This phenomenon can be observed clearly in CLOVER channel statistics.) It was during transits of the MUF through our operating frequencies that we logged most of the failed transfers that figured in our success-probability calculations.

Tables 3 and 4 show that the uncompressed over-the-air average throughput of CLOVER 2000 for text files on our links is around 91 cps on NVIS paths and 117 cps on OHS paths for daytime operations. These are about four times the corresponding CLOVER II throughputs. Nighttime NVIS and OHS throughput averages for uncompressed files are 59 and 83 cps, three to four times the corresponding CLOVER II throughputs. (The standard deviations of these average throughputs are all under 3 cps, making us confident that additional measurements made under the same conditions would yield similar mean throughputs.)

Average CLOVER 2000 throughputs for compressed text files are 166 and 181 cps on our NVIS and OHS paths during the day, and 93 and 164 cps at night, with standard deviations of mean throughput 5 cps or less. Compressed throughputs are again somewhat less than twice the corresponding uncompressed ones.

For the CLOVER 2000 transfers, average OHS throughputs are 20 or more percent larger than average NVIS throughputs, confirming expectations. This is no doubt the result of running the CLOVER 2000 NVIS and OHS tests on frequencies that had similar amounts of interference and lightning-induced noise.

Except for uncompressed nighttime OHS transfers, the probability of successful file transfer with CLOVER 2000 was 88% or higher (with the default maximum number of retries again set at 20). Uncompressed nighttime OHS transfers were successful 83% of the time. With a sample size of 113, this may merely reflect statistical variation. If not, it may be related to higher signal-to-noise-ratio requirements for the 2K-bandwidth CLOVER 2000 modulation than for the CLOVER II modulation, which had a somewhat higher transfer percentage (88%) in this case. An ALE system would probably also raise transfer success probabilities with CLOVER 2000.

The relatively large standard deviations of uncompressed and compressed file throughput for CLOVER II and 2000 (the standard deviations range from about ten to several tens of cps for one to two hundred transfers) reflect, in part, CLOVER's ability to adapt itself to changing channel conditions (see Sec. 1). However, these standard deviations are also affected by variability of file size, so channel adaptation should not be viewed as the sole source of throughput spread.

8. Comparison with PacTOR, GTOR and the HAL P38

In contrast to the daytime CLOVER II and CLOVER 2000 throughputs in Tables 1-4, PacTOR (with Huffman compression on) and GTOR (with its built-in compression) achieved average throughputs for text files of about 18 and 24 cps on our daytime NVIS links and 20 and 32 cps over daytime OHS links (data taken two or three years ago; see Ref. 4. We did not collect enough PacTOR, GTOR or P38 data for a comparison with their nighttime throughput.) Corresponding uncompressed throughputs would be about half of these values. Thus, if data compression is viewed as an intrinsic part of what we have called (Ref. 4) the "common" implementation of PacTOR and GTOR, then the HAL GUI software in its PKWARE-compression mode produces about twice the average throughput of PacTOR and GTOR for text-file transfers. CLOVER 2000, with four times the bandwidth of PacTOR and GTOR, has six to seven times the uncompressed throughput of PacTOR and GTOR and eight to ten times the compressed throughput.

The P38 achieved daytime NVIS throughputs for uncompressed and compressed text files of 24 and 43 cps. These are 20-30% lower than corresponding CLOVER II throughputs, as might be expected. Daytime P38 OHS throughputs were 29 and 50 cps for uncompressed and compressed text files, about 25% *higher* than the corresponding CLOVER II OHS throughputs. The reasons for this otherwise surprising result are the same as those for the reverse ordering of CLOVER II NVIS and OHS throughputs: unavoidably high levels of noise and interference during the CLOVER II OHS tests that

were not present during the CLOVER II and P38 NVIS tests. Against similar amounts of noise and interference, P38 NVIS throughputs would have been well below CLOVER II OHS throughputs. P38 maximum throughputs were about half those of CLOVER II.

9. CLOVER and File Encryption

To see if CLOVER file compression interferes with off-line file encryption (of interest in certain military and commercial applications), we encrypted and compressed several large (10-40k) text files with the commercial, Fortezza-based, LJLCryptoLib "LJLCryptor" application and sent them in compressed binary mode over a non-amateur CLOVER II NVIS link. Although the HAL GUI's PKWARE compression attempt produced, as expected, slightly larger versions of the already compressed and encrypted files, LJL decryption of the results restored the files to their original form as uncompressed text. CLOVER binary transfers thus appear to pass files encrypted by this approach transparently.

10. Concluding Remarks

We hope that our data will aid understanding of CLOVER file transfer over HF and perhaps serve as a useful introduction to how CLOVER works. CLOVER is now used all over the world by amateurs (in particular, for BBS mail forwarding). A number of international aid and other communications-providing organizations also use CLOVER to send information across parts of Africa and Australia, where alternative means of long-haul communication are not available, or too expensive. CLOVER modems are also used for at least two special-purpose e-mail systems (one used by private boating and shipping operators). Our data may shed light on why this inexpensive, amateur-developed modem and its protocols have become so popular.

Acknowledgments

References

1. Ken Wickwire (KB1JY), "A Software Package for Analyzing CLOVER Channel Statistics." *QEX,* November 1997.

2. Ken Wickwire (KB1JY) et al., "On-air Measurements of HF TOR and Packet Throughput, Part I: Near-Vertical-Incidence-Skywave Paths," *Digital Journal*, March 1996.

3. Ken Wickwire (KB1JY), "On-air Measurements of HF TOR and Packet Throughput, Part II: One-hop Skywave Paths," *QEX,* June 1996.

4. Ken Wickwire (KB1JY), "On-air Measurements of HF Data Throughput: Results and Reflections," *Proc. 15th ARRL/TAPR Digital Communications Conference*, ARRL, 1996.

5. Ken Wickwire (KB1JY) et al., "On-air Measurements of CLOVER P38 Throughput," *Proc. 16th ARRL/TAPR Digital Communications Conference*, ARRL, 1997.

Email: kwick@mitre.org. Packet radio: KB1JY@W1ON.

Notes

were not present during the CLOVER-II and P38 NVIS tests. Against similar amounts of noise and interference, P38 NVIS throughputs would have been well below CLOVER-II OHS throughputs. P38 groundwave throughputs were about half those of CLOVER-II.

9. CLOVER and File Encryption

To see if CLOVER file transmission interferes with off-line file encryption (of interest in certain military and commercial applications), we encrypted and compressed several large (10-400 kbyte) files with the commercial, Fortezza-based LJL Crypto[?] (b "LJLCrypto") application and sent them in compressed binary mode over a non-amateur CLOVER-II NVIS link through the Hal/GL's PKWARE compression, analysis produced, as expected, slightly larger versions of the already compressed and encrypted files. LJL decryption of the results returned the files to their original form as uncompressed text. CLOVER binary transfers thus appear to pass files encrypted by this approach transparently.

10. Concluding Remarks

We hope that our data will aid understanding of CLOVER file transfer over HF and perhaps serve as a useful introduction to how CLOVER works. CLOVER is now used all over the world by amateurs (in particular, for HF mail forwarding). A number of international aid and relief agencies and time-pressing organizations also use CLOVER to send information to and from parts of Africa and Australia where alternative means of long-haul communications are not available, in some instances, CLOVER stations are also used (in at least two special purpose naval systems for automated fleming and shipping operations). Our data may shed light on why this inexpensive, complex modem and its protocols have become so popular.

References

1. Ken Wickwire (KB1JY), "A Software Package for Analyzing CLOVER Channel Statistics," QEX, November 19??.

2. Ken Wickwire (KB1JY) et al., "On-air Measurements of HF CLOVER Packet Throughput, Part I: Near-Vertical-Incidence Skywave Paths," Digital Journal, March 1996.

3. Ken Wickwire (KB1JY), "On-air Measurements of HF TOR and Packet Throughput, Part II: Groundwave Paths," QEX, June 1996.

4. Ken Wickwire (KB1JY), "On-air Measurements of HF Data Throughput: Results and Reflections," Proc. 15th ARRL/TAPR Digital Communications Conference, ARRL, 1996.

5. Ken Wickwire (KB1JY) et al., "On-air Measurements of CLOVER P38 Throughput," Proc. 16th ARRL/TAPR Digital Communications Conference, ARRL, 1997.

Email: Wickwire@mitre.org. Packet radio: KB1JY @ W1MX.